NEDO 次世代ロボット知能化技術開発プロジェクト

自律移動システムマニュアル

1.0 版

豊橋技術科学大学

株式会社セック

改版履歴

版数	改版日	改版内容	備考
1.0	2012/02/29	初版作成	

目次

表目次

図目次

1　総則

1.1.　目的

本書は、自律移動システムの使用方法について、記述した文書である。

1.2.　適用範囲

本書は、自律移動モジュール群を用いた自律移動システムに対して適用する。

1.3.　関連文書等

1.3.1.　適用文書

なし

1.3.2.　関連文書

表 1-1　関連文書一覧

No	文書名	版数	発行元
1	自律移動モジュール群マニュアル	1.0	独立行政法人産業技術総合研究所
2	Bumblebee2Module コンポーネント取扱説明書	平成 23 年 7 月 30 日	豊橋技術科学大学 行動知能システム学研究室
3	URG センサ RT コンポーネントマニュアル	1.8	株式会社セック
4	センサ RTC 共通マニュアル	1.7	株式会社セック
5	SCIP2.0 準拠“URG”シリーズ通信仕様書	最新版	北陽電機株式会社
6	人物検出コンポーネント	平成 23 年 7 月 19 日	豊橋技術科学大学 行動知能システム学研究室
7	自己位置推定コンポーネント取扱説明書	平成 23 年 10 月 27 日	豊橋技術科学大学 行動知能システム学研究室
8	SLAM コンポーネント取扱説明書	平成 23 年 10 月 27 日	豊橋技術科学大学 行動知能システム学研究室
9	LocalMap コンポーネント取扱説明書	平成 23 年 7 月 19 日	豊橋技術科学大学 行動知能システム学研究室
10	環境シミュレータ RTC 取扱説明書	平成 23 年 11 月 7 日	豊橋技術科学大学 行動知能システム学研究室
11	経路計画コンポーネント	平成 23 年 7 月 21 日	豊橋技術科学大学 行動知能システム学研究室
12	MobileRobots 社ロボット用制御コンポーネント	平成 23 年 7 月 16 日	豊橋技術科学大学 行動知能システム学研究室
13	大域経路計画コンポーネント	平成 24 年 1 月 11 日	豊橋技術科学大学 行動知能システム学研究室

1.3.3.　参考文書

表 1-2　参考文書一覧

No	文書名	版数	発行元
1	【Web】行動知能システム学研究室 http://www.aisl.cs.tut.ac.jp/ http://www.aisl.cs.tut.ac.jp/RTC/index.html	―	豊橋技術科学大学

1.4.　定義

1.4.1.　用語

表 1-3　自律移動システム　用語一覧

No	用語	説明
1	RTM	RT ミドルウェア
2	RTC	RT コンポーネント
3	CORBA	分散オブジェクト技術の仕様
4	ステレオ距離画像	左右に配置した２台のカメラで撮影した２枚一組の画像に写った物体等の対応点を比較し、対応する点のずれ（視差）をもとに、対象物までの距離を算出した距離データ付きの画像情報。
5	カメラキャリブレーション	カメラの位置・向きといったカメラ外部のパラメータや、焦点距離、CCD の画素の縦横費といった内部的なパラメータをもとめ、カメラを使用可能な状態にすること。これにより、二次元画像と三次元空間の点との対応関係を求めることができる。
6	大域地図	距離データやロボットの移動量などをもとに作成した、ロボット周囲の障害物存在確立地図のこと。PNG などの画像データ形式で作成される。局所地図に比べ、比較的広い範囲で作成され、ロボットの経路計画の作成や、現実世界の自己位置の推定、壁やドアのような静的な障害物を検出に利用することができる。
7	局所地図	距離データやロボットの移動量などをもとに作成した、ロボット周囲の障害物存在確立地図のこと。自律移動モジュール群では、局所地図をリアルタイムで作成、使用することで、人物といった動的な障害物を検知することが可能となる。
8	レーザ距離センサ	レーザ光線により、対象物との距離計測を行うことができるセンサ。自律移動モジュール群が使用する Top-URG センサは、半円状のフィールドをスキャンし、対象物との距離を角度ごとに取得することができる測域センサである。
9	オドメトリ	ロボットの走行距離を制御するためには、自己位置の推定が必要となる。そのための手法の一つで、タイヤの車輪やステアリングの回転角度から移動量を求め、ロボットの事故位置を推定する方法の総称。または、それにより得られた情報のこと。

No	用語	説明
10	SCIP2.0	測域センサコマンドインタフェース研究会（筑波大学知能ロボット研究室(http://www.roboken.esys.tsukuba.ac.jp)が制定した、測域センサとの通信プロトコル。URGセンサーコンポーネントが通信するClassic-URG、Top-URGはともにSCIP2.0 に対応しており、各センサとの通信で使用している。
11	ステップ番号	URG センサにおいて、角度を示す値。
12	まとめるステップ数	URG センサに設定する値。 URG センサでは、角度毎の距離データを出力する。本ステップ数分のデータがまとめられる。センサの規定の角度解像度ではなく、より大きい角度解像度のデータとしたい場合に使用する。 例えば、200 ステップ分の距離データを要求し、まとめるステップ数として"02"を設定した場合、100 ステップ分の距離データが出力される。まとめられたデータは、各ステップグループ中の最小値となる。
13	間引きスキャン数	URG センサに設定する値。本スキャン数おきに距離を出力する。 例えば'3'が指定された場合、1 スキャンの計測・距離出力後 3 スキャンを間に挟んで 1 スキャンの計測・送信を行う。

1.4.2.　座標系

コンポーネントが使用している座標系について記述する。

(1)　レーザ距離データの座標系

レーザ距離センサから距離データを取得するための座標系。

- 原点：右方向を基準（0°）とする。
- 計測開始位置（degree）、計測終了位置（degree）を指定し、計測データの間隔（degree）に応じて、各方向の物体までの距離（mm）を計測することができる。

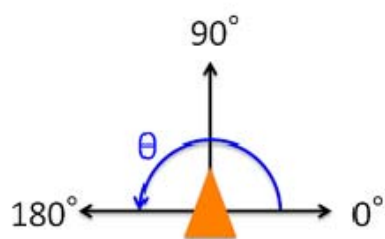



図 1-1　レーザ距離データの座標系

(2)　ロボット位置・姿勢の座標系

ロボットの位置・姿勢を取得するための座標系。

・　x：原点座標から X 軸方向の距離（m）
・　y：原点座標から Y 軸方向の距離（m）
・　heading：X 軸方向を原点としたロボットの向き（radian）

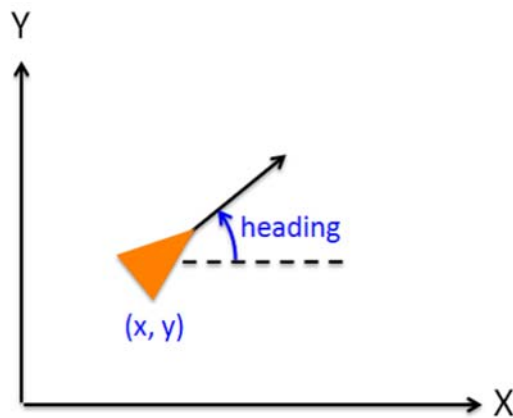



図 1-2　ロボット位置・姿勢の座標系

(3)　大域地図の座標系

大域地図の座標系は、大きさやスケールといった情報で表現される。
なお、大域地図を扱う場合の origin の姿勢情報には常に 0 となる。

- xScale : X 軸方向の地図のスケール（m/cell）
- yScale : Y 軸方向の地図のスケール（m/cell）
- width : X 軸方向の地図の大きさ（cell）
- height : Y 軸方向の地図の大きさ（cell）
- origin : ロボット中心から見た cell(0,0) の絶対座標

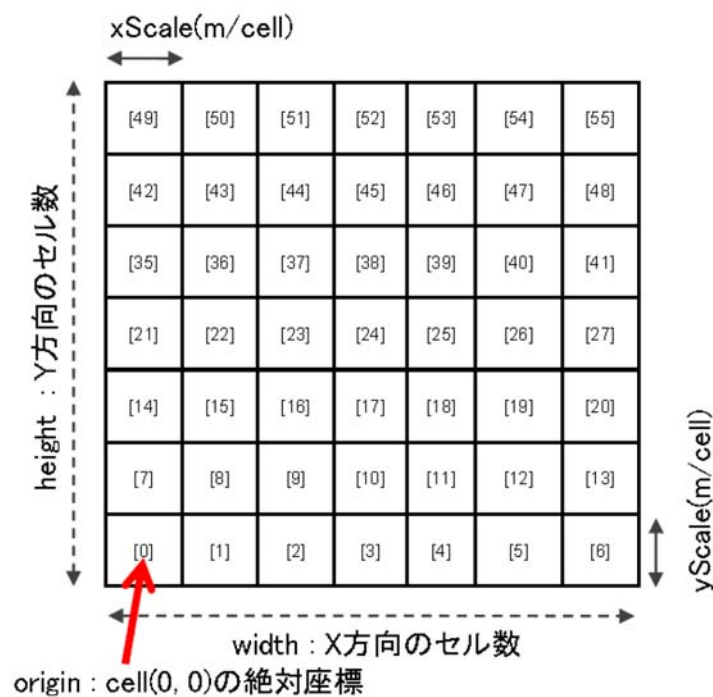



図 1-3　大域地図の座標系

(4)　局所地図の座標系

局所地図の座標系は、大きさやスケールといった情報で表現される。

- xScale：X 軸方向の地図のスケール（m/cell）
- yScale：Y 軸方向の地図のスケール（m/cell）
- width：X 軸方向の地図の大きさ（cell）
- height：Y 軸方向の地図の大きさ（cell）
- origin：ロボット中心から見た cell(0,0) の絶対座標
- pose：ロボット中心の絶対座標

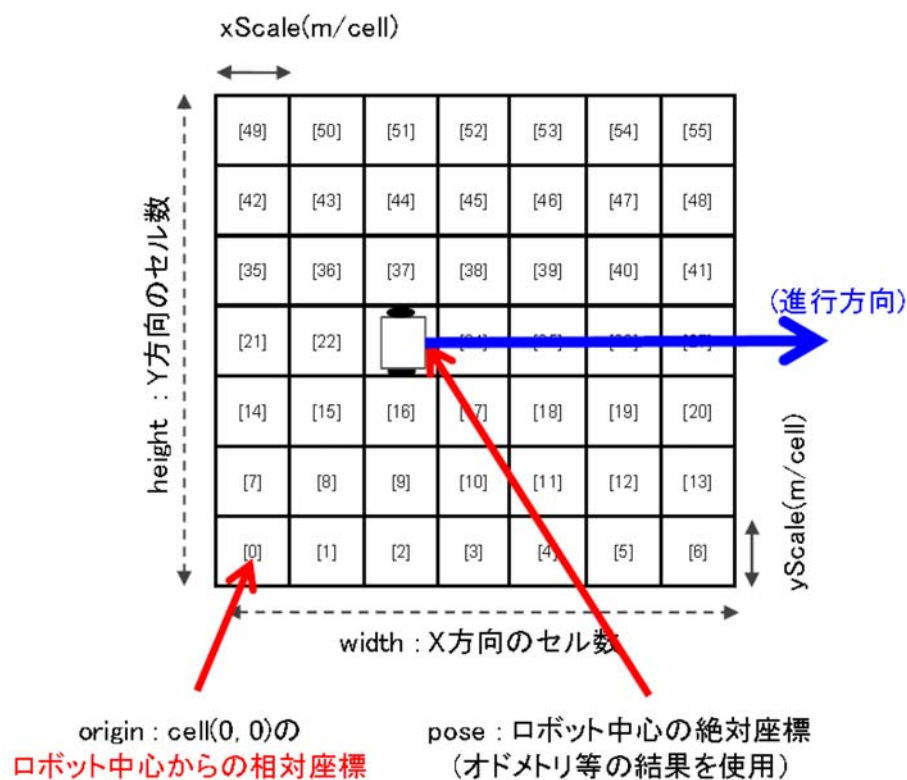



図 1-4　局所地図の座標系

1.5.　ライセンス

1.5.1.　自律移動モジュール群

自律移動モジュール群のうち、豊橋技術科学大学が作成した RTC のライセンスは、修正 BSD ライセンスに従う。

Top-URG センサ RTC のライセンスは、株式会社セック(以降、権利者)が所有している。

Top-URG センサ RTC、並びに、これに使用するサンプル RTC、操作説明等のドキュメント(以降、Top-URG センサ RTC 等)の利用は、以下の条件に同意した個人、またはグループ(以降、利用者)にのみ許諾されるものとする。

(1) 権利者は、Top-URG センサ RTC 等の利用、利用不能、サポートサービスの提供、サポートサービスの不提供により利用者に生じる一切の損害に関して、一切の責任を負わない。たとえ、権利者がこのような損害発生の可能性について事前に知らされていた場合でも同様。

(2) Top-URG センサ RTC 等のリバースエンジニアリング（調査・解析を行い、プログラム構造などの技術を探知する行為）を禁止する。

1.5.2.　使用ツール・ライブラリ

自律移動モジュール群が内部で用いているソフトウェアは、各々のライセンスに従う。

(1)　Open-rtm-aist 1.0.0 (C++版)

Open-rtm-aist 1.0.0 (C++版)は、EPL (Eclipse Public License) ライセンス、または産業技術総合研究所（AIST）との個別契約のうち、一つから選択するデュアルライセンス方式で利用することができる。

(2)　omniORB

omniORB は、GPLv2（GNU General Public License v2）、LGPLv2（GNU Lesser General Public License v2）の下で自由に利用することができる。

詳細は omniORB の公式サイトを参照。

http://omniorb.sourceforge.net/

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(3)　OpenCV

OpenCV およびそのソースコードのライセンスは、BSD ライセンスに従う。

詳細は OpenCV の公式サイトを参照。

http://opencv.willowgarage.com/wiki/

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(4)　FlyCapture、Triclops

FlyCapture、Triclops の権利は、PointGrey 社が所有している。

FlyCapture のライセンスは以下の URL に記載された条項（英文）に従う。

http://www.ptgrey.com/support/kb/data/eula.rtf

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

Triclops のライセンスは、ダウンロードした Triclops に同梱される、「Triclops Software Development Kit(SDK) Manual Reference（英文）」に記載されるライセンス条項に従う。

Triclops は、PointGrey 社のサイトよりダウンロードできる。

http://www.ptgrey.com/index.asp

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(5)　ARIA

ARIA およびそのソースコードのライセンスは、GPL（GNU General Public License）に従う。ただし、独自のディストリビューション（例えば、独自のソースコードを公開しない場合）は商用ライセンスが必要。

詳細は、MobileRobots 社のサポートサイト(英文)を参照。

http://robots.mobilerobots.com/wiki/ARIA

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(6)　CGAL

CGAL のライセンスデュアルライセンス方式となっており、オープンソフトウェアで使用する場合は、LGPL（GNU Lesser General Public License）、もしくは、QPL（Q Public License）のもとで使用することが可能。

詳細は、CGAL のサイト(英文)を参照。

http://www.cgal.org/license.html

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

2　システム構成

2.1.　システム概要

自律移動システムは、ステレオカメラ、レーザ距離センサ、ロボットなどのハードウェアを制御する RTC や、それらの RTC から得られた様々な情報を利用して経路計画を作成する RTC、人物検出を行う RTC といった機能を持つ自律移動モジュール群から構成される。

これら、ロボットを構成する要素となる RTC を組み合わせて利用することで、自律移動を行うロボットを容易に構築することが可能になる。

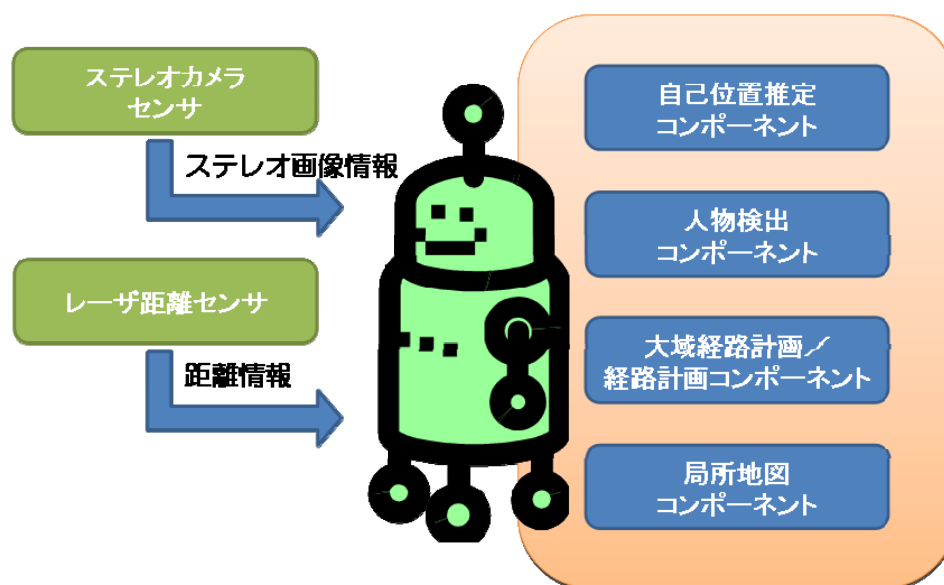



図 2-1　自律移動システム

自律移動システムは、以下の自律移動モジュール群を中心に構成される。

表 2-1　自律移動モジュール群コンポーネント

コンポーネント	説明	RTC 名	依存ライブラリ（バージョン）
ShowImage コンポーネント	カメラ画像など画像情報を表示するためのコンポーネント	ShowImageComp	OpenCV（2.1）
Top-URG センサ RTC	北陽電機 URG センサから距離データを取得するためのコンポーネント	URGDataFlowComp Comp	－
人物検出 RTC	ステレオカメラ（Bumblebee2）の情報から人物を検出し、それぞれの人物の位置と移動速度を出力するコンポーネント	PeopleTrackingV2Co mp	OpenCV（2.1） FlyCapture（1.7） Triclops（3.2）
ロボット自己位	データポートから入力された	LocalizationComp	OpenCV（2.1）

コンポーネント	説明	RTC 名	依存ライブラリ（バージョン）
置推定 RTC	距離データとロボットの移動量を用いて大域地図上でのロボット自己位置を推定するコンポーネント	SimpleGlobalMapLoaderComp	OpenCV（2.1）
大域地図生成・表示 RTC	独自手法の大域地図の生成・表示を行うコンポーネント群	SLAMComp	OpenCV（2.1）
局所地図生成・更新 RTC	距離データとロボットの移動量よりロボット周囲の障害物存在確率マップを作成するコンポーネント	LocalMapComp	―
移動ロボットのソフトウェア開発のための屋内環境シミュレータ RTC	屋内環境で行動するロボット、人の動きを再現し、センサ出力や人物検出結果、地図などさまざまな情報を出力する。また、ロボット制御命令を入力することによってシミュレータ上の仮想移動ロボットを制御することができるコンポーネント	EnvironmentSimulatorComp	OpenCV（2.1） CGAL（3.7）
経路計画 RTC	静止・移動障害物を回避しながら、指定物体を追う経路を計画するコンポーネント	PathPlannerV2Comp	OpenCV（2.1）
MobileRobots 社ロボット用制御 RTC	MobileRobots 社のロボット用の制御コンポーネント	MobileRobotsControllerComp	ARIA（2.7.1）
		Dumy_velocity_dataComp	―
大域経路計画 RTC	大域地図の情報を基にロボットが移動する大域的な経路を計画するコンポーネント	GlobalPathPlanner	―
		Dummy2PosesSenderComp	―

2.2. 使用条件

システムの動作環境を表 2-2に示す。

表 2-2 動作環境

No.	要求環境			備考
1	OS	Windows	WindowsXP SP3	
2	ミドルウェア	OpenRTM-aist	1.0.0-RELEASE （C++)	
3		omniORB	4.1.4	
4	ツール	RT-SystemEditor		RTC の操作に必要となる

2.3. ハードウェア仕様

自律移動システムのためのハードウェアの一覧を表 2-3に示す。

表 2-3 自律移動システム ハードウェア一覧

No	種別	メーカー	型番	説明
1	Bumblebee2	PointGrey 社	BB2-08S2C-25	ステレオカメラ （XGA、カラー、画角 110 度）
2	Top-URG	北陽電気	UTM-30LX	レーザ式測域センサ
3	PeopleBot	MobileRobots 社	－	ロボット
4	Pioneer3	MobileRobots 社	－	ロボット

3　サービス仕様

3.1.　サービス一覧

自律移動モジュール群を利用した自律移動システムのサービスを表 3-1に記述する。

表 3-1　自律移動システム　サービス一覧

No	名称
1	移動ロボット用環境シミュレータ
2	自律移動ロボットによる人物追従

3.2.　移動ロボット用環境シミュレータ

環境シミュレータ RTC は、大学校舎内や学生食堂といったある程度広い屋内環境を再現し、様々な RTC の動作を検証することができる環境を提供する。

また、環境内を移動する多数人物の行動を再現することにより、安全に経路計画などのアルゴリズムを検証することができる。

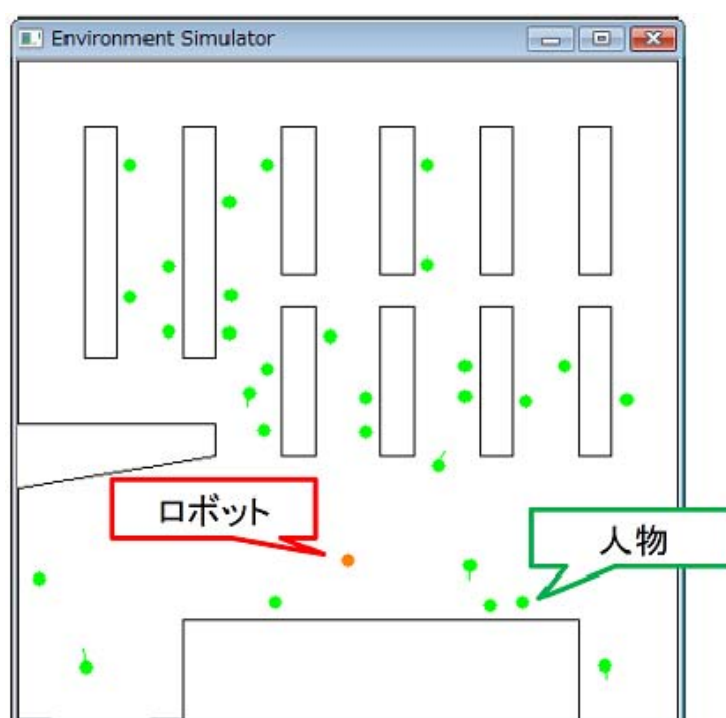



図 3-1　環境シミュレータ RTC：起動例

環境シミュレータ RTC は、以下の特徴を持つ。

・　ロボット位置、レーザ距離情報、地図情報、人物情報などをシミュレータから取得できる。
・　環境内の多数の人の動きをシミュレーションすることができる。

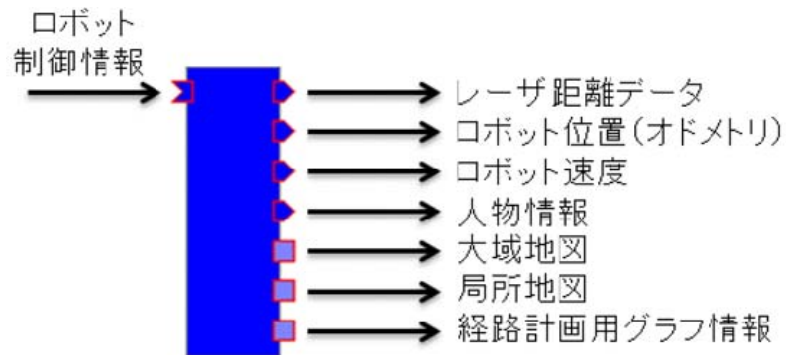



図 3-2　環境シミュレータ RTC : インタフェース

3.2.1.　サービス内容

環境シミュレータ RTC は、他の自律移動モジュール群と組み合わせて利用し、それらのモジュールのテストや評価を行う。以下に、他のモジュールと組み合わせて利用した例を以下に示す。

(1)　人物追従

環境シミュレータ RTC と経路計画 RTC を組み合わせ、シミュレータ上の仮想ロボットを制御し、静止・移動障害物を回避しながら、特定の人物を追従することが可能である。

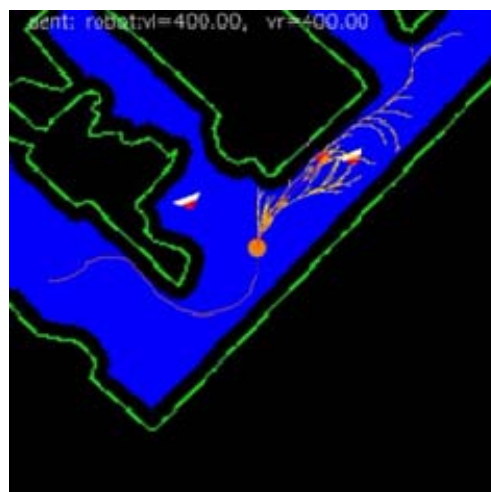



図 3-3　人物追従 : 実行例

(2) 大域地図生成

環境シミュレータ RTC と大域地図生成 RTC を組み合わせ、シミュレータ上で仮想ロボットを制御し、得られたレーザ距離データより、大域地図を生成する。

仮想ロボットは、Dummy_velocity_data コンポーネントにロボット制御情報（並進速度[m/s]、角速度[rad/s]）を入力することで制御することができる。

並進速度、加速度は、半角スペースで区切って入力する。

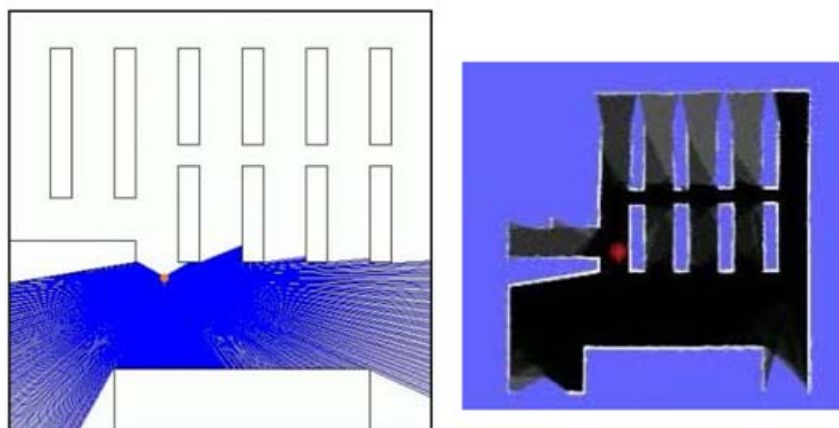



図 3-4　大域地図生成：実行例

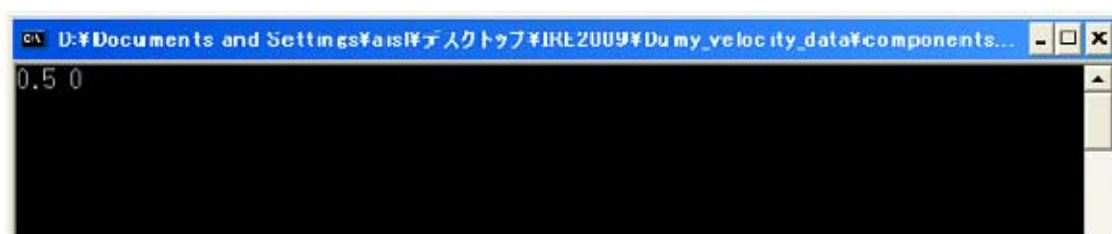



図 3-5　Dummy_velocity_data コンポーネント：データ入力例

(3) 大域経路計画と自律移動

環境シミュレータ RTC と大域経路計画 RTC、経路計画 RTC を組み合わせ、シミュレータ上で、仮想ロボットを自律移動させることができる。

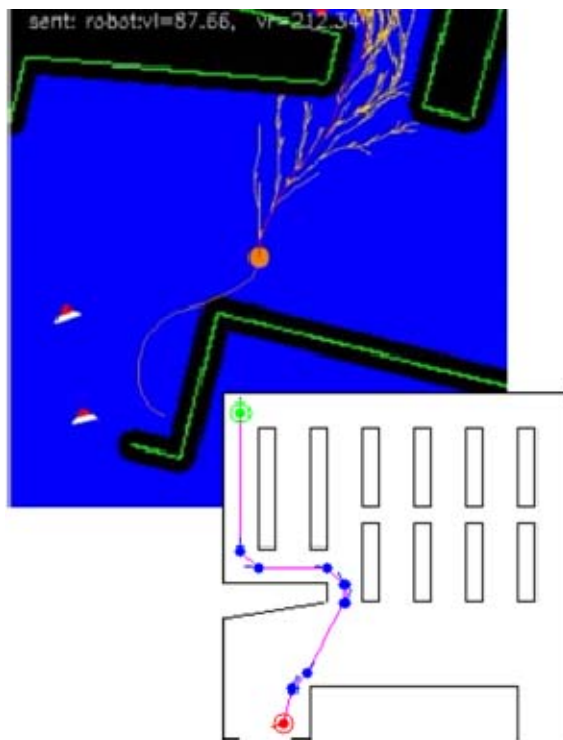



図 3-6　大域経路計画と自律移動：実行例

仮想ロボットが自律移動するための開始位置、終了位置は、Dummy2PosesSender コンポーネントにより与えられ、当該コンポーネントがアクティブ化する時に、関連コンポーネントへ出力される。

3.2.2. 動作条件・制約

環境シミュレータ RTC が利用する環境データは、テキストファイルとして用意する。

既に、簡易データ、学食、スーパーマーケットをモデル化したテキストファイルが用意されているが、新規に作成することもできる。

環境データの詳細は、「4.2.1 (2)」を参照。

(1) 人物追従

シミュレータ上の人物を追従するため、経路計画 RTC（PathPlannerV2 コンポーネント）のコンフィギュレーションを以下のように設定する必要がある。

- USE_PEOPLE_TRACKING　：1（人物追跡を使う）
- USE_WAYPOINT_PORT　：0（経由点は使わず人物追跡データを使用する）

コンフィギュレーションの詳細は、「6.1 (1)」を参照のこと。

(2) 大域地図生成

特になし。

(3) 大域経路計画と自律移動

特になし。

3.2.3. システム構成

(1) 人物追従

人物追従のシステム構成図を図 3-7に、使用RTCを表 3-2に示す。

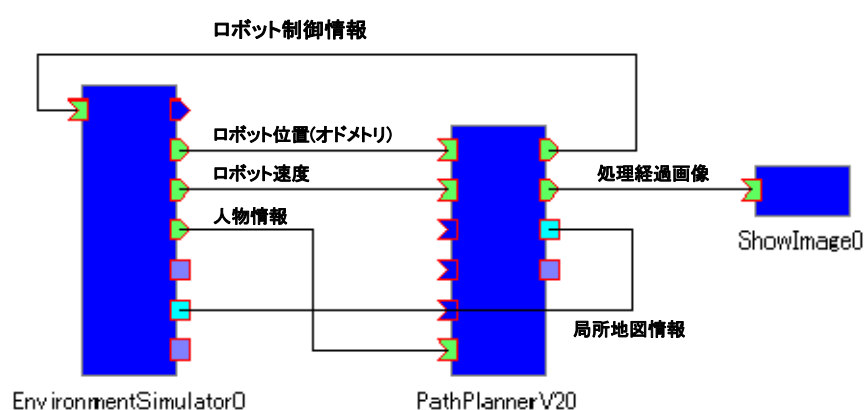



図 3-7　人物追従：システム構成

表 3-2　人物追従：使用 RTC 一覧

No	コンポーネント	RTC 名	数
1	環境シミュレータ RTC	Simulator／EnvironmentSimulatorComp	1
2	経路計画 RTC	PathPlanner v2.6 ver.IIS／PathPlannerV2Comp	1
3	ShowImage コンポーネント	Bamblebee2Module／ShowImageComp	1

(2) 大域地図生成

大域地図生成のシステム構成図を図 3-8に、使用RTCを表 3-3に示す。

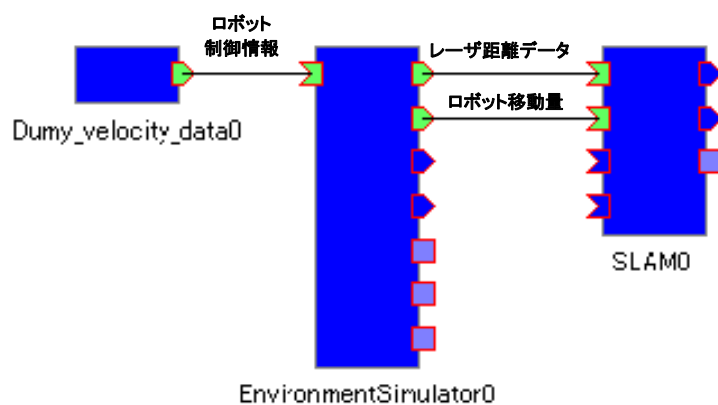



図 3-8　大域地図生成：システム構成

表 3-3　大域地図生成：使用 RTC 一覧

No	コンポーネント	RTC 名	数
1	環境シミュレータ RTC	Simulator／EnvironmentSimulatorComp	1
2	大域地図生成 RTC	SLAM／SLAMComp	1
3	Dummy_velocity_data コンポーネント	MobileRobotsController／Dumy_velocity_dataComp	1

(3) 大域経路計画と自律移動

大域経路計画と自律移動のシステム構成図を図 3-9に、使用RTCを表 3-4に示す。

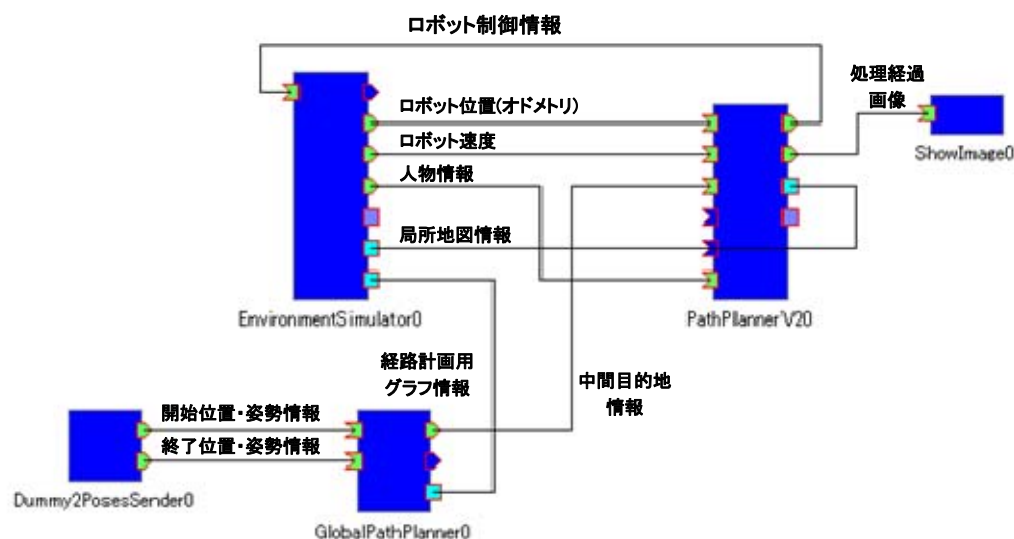



図 3-9　大域経路計画と自律移動：システム構成

表 3-4　大域経路計画と自律移動：使用 RTC 一覧

No	コンポーネント	RTC 名	数
1	環境シミュレータ RTC	Simulator／EnvironmentSimulatorComp	1
2	経路計画 RTC	PathPlanner v2.6 ver.IIS／ PathPlannerV2Comp	1
3	ShowImage コンポーネント	Bamblebee2Module／ShowImageComp	1
4	大域経路計画 RTC	GlobalPathPlanner／ GlobalPathPlannerComp	1
5	Dummy2PosesSender コンポーネント	GlobalPathPlanner／ Dummy2PosesSenderComp	1

3.3. 自律移動ロボットによる人物追従

様々な RTC を組み合わせることで、現実世界で人物追従を行う自律移動ロボットを構築することができる。

3.3.1. サービス内容

- Top-URG センサから得たレーザ距離データに基づく距離データの取得。
- ロボット位置情報（オドメトリ）、レーザ距離データに基づく局所地図の生成。
- MobileRobots 社ロボット用制御 RTC を利用したロボットの制御。
- ロボット位置情報（オドメトリ）、レーザ距離データに基づく、大域地図上の自己位置の推定。
- ステレオカメラ「Bumblebee2」取得したステレオ距離画像をもとに、人物情報の作成。
- ロボット制御情報、人物情報を基にした人物を追従する経路計画の作成。

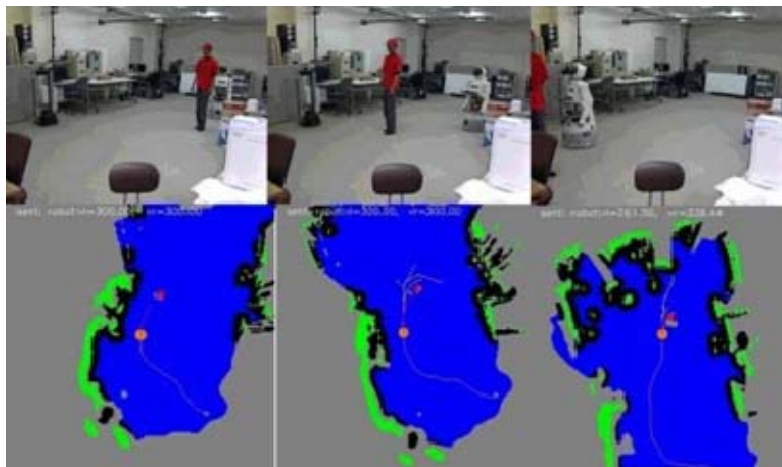



図 3-10　自律移動ロボットによる人物追従：実行例

3.3.2. 動作条件・制約

- 大域地図情報は事前に作成しておく必要がある。

3.3.3.　システム構成

自律移動ロボットによる人物追従のシステム構成を図 3-11に、使用RTCを表 3-5に示す。

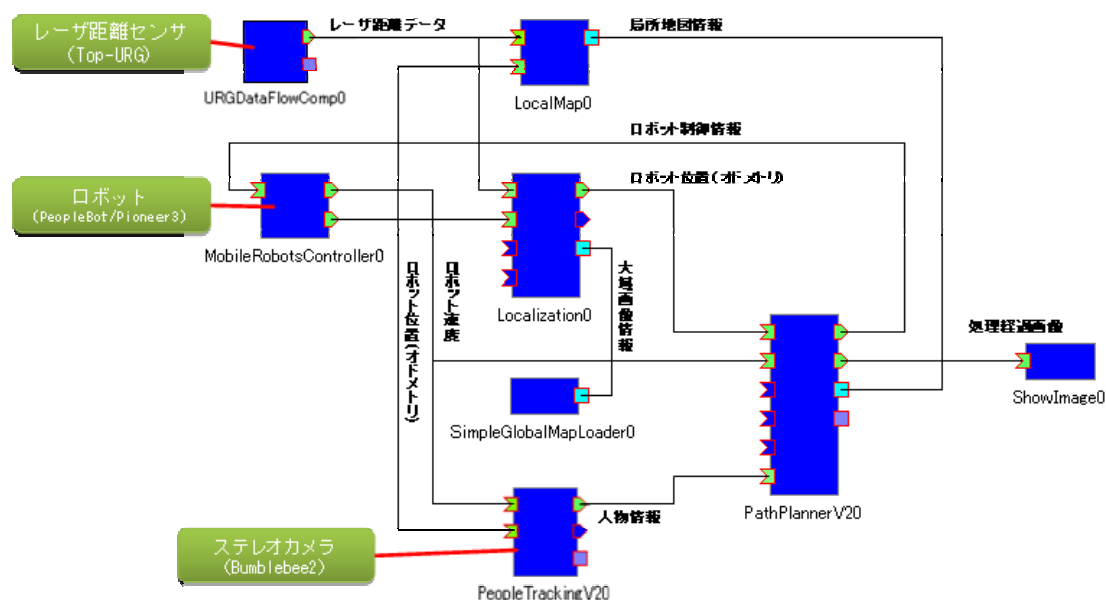



図 3-11　自律移動ロボットによる人物追従：システム構成

表 3-5　自律移動ロボットによる人物追従：使用 RTC 一覧

No	コンポーネント	RTC 名	数
1	Top-URG センサ RTC	TopURG_VCxx_exe／URGDataFlowCompComp	1
2	局所地図生成・更新 RTC	LocalMapRTC／LocalMapComp	1
3	MobileRobots 社ロボット用制御 RTC	MobileRobotsController／MobileRobotsControllerComp	1
4	ロボット自己位置推定 RTC	Localization／LocalizationComp	1
5	SimpleGlobalMapLoader コンポーネント	Localization／SimpleGlobalMapLoaderComp	1
6	経路計画 RTC	PathPlanner v2.6 ver.IIS／PathPlannerV2Comp	1
7	ShowImage コンポーネント	Bamblebee2Module／ShowImageComp	1
8	人物検出 RTC	PeopleTrackingV2／PeopleTrackingV2Comp	1

4　取扱手順

Windows 環境における自律移動モジュール群の取り扱い手順について記述する。

4.1.　環境構築

自律移動モジュール群を動作させるために、以下のソフトウェアパッケージをインストールする。各コンポーネントとの関係を表 4-1に示す。

- Open-rtm-aist 1.0.0(C++版)
- omniORB 4.1.4
- OpenCV 2.1
- FlyCapture 1.8
- Triclops 3.2
- ARIA 2.7.1
- CGAL 3.7

表 4-1　コンポーネントとソフトウェアパッケージの関係

No.	コンポーネント	Open-rtm-aist 1.0.0	omniORB	OpenCV 2.1	FlyCapture 1.8	Triclops 3.2	ARIA 2.7.1	CGAL 3.7
1	ShowImageComp	○	○	○				
2	URGDataFlowComp	○	○					
3	PeopleTrackingV2Comp	○	○	○	○	○		
4	LocalizationComp	○	○	○				
5	SimpleGlobalMapLoaderComp	○	○	○				
6	SLAMComp	○	○	○				
7	LocalMapComp	○	○					
8	EnvironmentSimulatorComp	○	○	○				○
9	PathPlannerV2Comp	○	○	○				
10	MotionSet_setting	○	○					
11	MobileRobotsControllerComp	○	○				○	
12	Dumy_velocity_dataComp	○	○					
13	GlobalPathPlanner	○	○					
14	Dummy2PosesSenderComp	○	○					

4.1.1.　インストール

(1)　Open-rtm-aist 1.0.0(C++版)

(a)　Open-rtm-aist 1.0.0(C++版)のインストール

OpenRTM-aistはロボットシステムをコンポーネント指向開発するためのソフトウェアプラットフォームである。以下のURLよりインストーラをダウンロードしインストールする。インストールの詳細は表 4-2の「インストール方法の解説」を参照のこと。

表 4-2　OpenRTM-aistのダウンロードURL

ダウンロードページ	http://www.openrtm.org/openrtm/ja/content/openrtm-aist-100-release#toc2
ダウンロードファイル	OpenRTM-aist-1.0.0-RELEASE_vc9_100212.msi
インストール方法の解説	http://www.openrtm.org/openrtm/ja/content/windowsへのインストール

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(b)　JREのインストール

OpenRTM-aistのToolであるRT System Editorを動かすためには、JavaのJRE (Java Runtime Environment)またはJDK (Java Development Kit)が必要である。以下のURLよりインストーラをダウンロードし、必要なパッケージをインストールする。インストールの詳細は表 4-3の「インストール方法の解説」を参照のこと。

表 4-3　JREのダウンロードURL

ダウンロードページ	http://www.java.com/ja/download/manual.jsp
ダウンロードファイル	http://javadl.sun.com/webapps/download/AutoDL?BundleId=58139
インストール方法の解説	http://www.java.com/ja/download/help/windows_manual_download.xml

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(2)　omniORB 4.1.4

omniORB は OpenRTM-aist1.0.0 に同梱されている。個別のインストールは必要ない(※)。

（※）Open-rtm-aist 1.0.0(C++版)をカスタムモードでインストールすると、omniORBpy がインストールされない場合があります。omniORB がインストールされているか確認してください。

(3)　OpenCV2.1

OpenCVは画像処理のためのソフトウェアパッケージである。コンポーネントにより必要とするOpenCVのバージョンが異なるため、それぞれ以下のURLよりインストーラをダウンロードし、必要なパッケージをインストールする。インストールの詳細は表 4-4の「インストール方法の解説」を参照のこと。

表 4-4　OpenCV2.1 のダウンロードURL

ダウンロードページ	http://sourceforge.net/projects/opencvlibrary/files/opencv-win/
ダウンロードファイル	OpenCV-2.1.0-win32-vs2008.exe
インストール方法の解説	http://www.java.com/ja/download/help/windows_manual_download.xml

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(4)　FlyCapture 1.8

FlyCaputre はステレオカメラから得られる画像にアクセスするためのライブラリである。以下の URL よりインストーラをダウンロードし、インストールする。

表 4-5　FlyCapture のダウンロード URL

ダウンロードページ	http://www.ptgrey.com/support/downloads/download.asp
ダウンロードファイル	flycapture_1_8_3_26_x86.exe
インストール方法の解説	―

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(5)　Triclops 3.2

Triclops はステレオカメラから得られる画像にアクセスするためのライブラリである。以下の URL よりインストーラをダウンロードし、インストールする。

表 4-6　Triclops 3.2 のダウンロード URL

ダウンロードページ	http://www.ptgrey.com/support/downloads/download.asp
ダウンロードファイル	triclops3.3b03_x86.exe
インストール方法の解説	―

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(6)　ARIA 2.7.1

ARIA は MobileRobots プラットフォームのライブラリである。以下の URL よりインストーラをダウンロードし、インストールする(※)。

表 4-7　ARIA 2.7.1 のダウンロード URL

ダウンロードページ	http://robots.mobilerobots.com/ARIA/download/archives/
ダウンロードファイル	ARIA-2.7.1.exe
インストール方法の解説	http://robots.mobilerobots.com/ARIA/INSTALL.txt

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(※) version 2.7.1 以外のライブラリではコンポーネントが起動しない可能性があります。

(7)　CGAL 3.7

CGAL は計算幾何学アルゴリズムの C++ライブラリである。

表 4-8　CGAL のダウンロード URL

ダウンロードページ	http://www.cgal.org/download.html
ダウンロードファイル	CGAL-3.7-Setup.exe
インストール方法の解説	－

[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

4.1.2.　動作確認環境の準備

(1)　RT SystemEditor

RT SystemEditor は、RTC をリアルタイムにグラフィカル操作する機能を持つ。OpenRTM-aist 1.0.0(C++版)に含まれる開発ツールの 1 つである。スタートメニューフォルダから「OpenRTM-aist」>>「C++」>>「tools」>>「RT SystemEditor」を選択し、RT SystemEditor を起動する。起動すると、図のような画面が表示される。

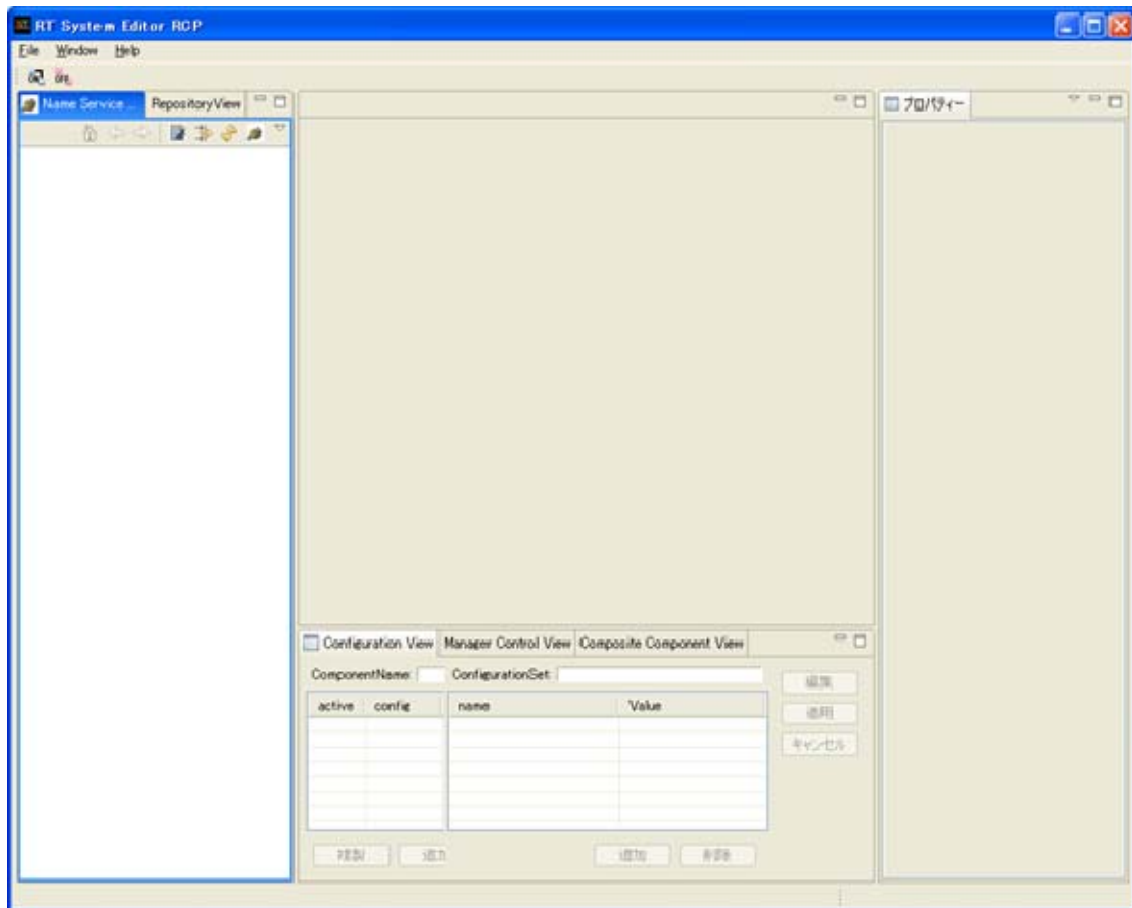



図 4-1　RT SystemEditor の起動

起動方法の詳細な手順については、以下の URL を参照のこと。

http://openrtm.org/openrtm/ja/content/%E5%8B%95%E4%BD%9C%E7%A2%BA%E8%AA%8D-windows%E7%B7%A8%23toc11#toc11
[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

(2)　rtc.confの作成

rtc.conf は、各コンポーネントのコンフィグレーション（ネームサービスやログ出力等）を設定するファイルである。コンフィギュレーションファイルは通常 rtc.conf という名前で作成されるが、任意の名前で作成したコンフィギュレーションファイルを渡すこともできる。各コンフィグレーションの設定内容については、以下の URL を参照のこと。

http://openrtm.org/openrtm/ja/content/%E5%8B%95%E4%BD%9C%E7%A2%BA%E8%AA%8D-windows%E7%B7%A8%23toc11%23toc11#toc11
[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

コンポーネント独自のコンフィギュレーションのパラメータ項目と取り得る値については、各コンポーネントの『コンフィギュレーション』の項を参照のこと。

(3)　ネームサーバの起動

ネームサーバは、コンポーネントの参照を登録するためのサーバである。スタートメニューフォルダから「OpenRTM-aist」>>「C++」>>「tools」>>「Start Naming Service」を選択し、ネームサーバを起動する。起動すると、図のようなコンソール画面が表示される。
（注意：コンソールを終了すると、RT SystemEditor 上でコンポーネントを参照することができなくなる）

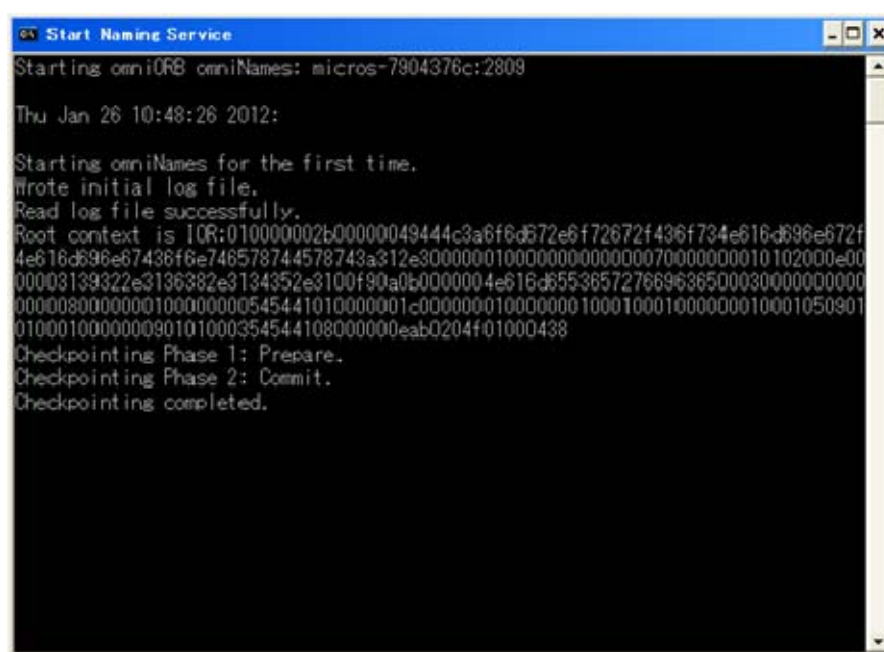



図 4-2　ネームサーバの起動

起動方法の詳細については、以下の URL を参照のこと。

http://openrtm.org/openrtm/ja/content/%E5%8B%95%E4%BD%9C%E7%A2%BA%E8%AA%8D-windows%E7%B7%A8%23toc11%23toc11%23toc11#toc2
[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

4.2.　設定・カスタマイズ手順

本システムは様々なコンポーネントを組み合わせて利用する。

環境シミュレータ RTC は、コンフィギュレーション、および、環境データのテキストファイルをカスタマイズすることで、様々な環境のシミュレーションが可能である。

また、経路計画コンポーネントは、使用するロボットに合わせて、ロボットの動作セットを定義するファイルが必要となる。

4.2.1.　環境シミュレータRTC

環境シミュレータ RTC は屋内環境、そこで行動するロボット、そして環境内の人の動きを再現する。歩行者の動きは「大域的な動き」および「個人レベルの局所的な動き」に分類され、シミュレータではポテンシャルモデルと移動ネットワークを組み合わせることによってこれを再現する。

また、障害物（壁）の情報と人の行動に関する情報が記録された環境データはテキストファイルとして用意する。

障害物の情報は 2 つの座標を結ぶ線分として表現されており、線分の集合として表現することができればある程度複雑な環境でも再現することができる。環境データのファイル名をシミュレータ RTC のコンフィグレーションで指定することにより、環境をカスタマイズしてシミュレーションを行うことができる。

(1)　コンフィギュレーション

環境シミュレータ RTC が持つコンフィグレーションについて説明する。

(a)　FieldDataFile

表 4-9　環境データを指定するコンフィグレーション

No.	型	変数名	単位
1	std::string	FieldDataFile	-

FieldDataFile でシミュレーションに用いる環境のデータを指定する。環境データには障害物（壁）の情報と人の行動に関する情報が記録されている。環境データはテキストファイルとして用意する。障害物の情報は 2 つの座標を結ぶ線分として表現されており、線分の集合として表現することができれば、ある程度複雑な環境でも再現することができる。環境データのファイル名をこのコンフィグレーションで指定することにより、簡単に環境を切り替えてシミュレーションを行うことができる。

この環境データの仕様については後述する。

(b) walkingrate_mean、walkingrate_std_deviation

表 4-10　人の歩行速度に関するコンフィグレーション

No.	型	変数名	単位
1	double	walkingrate_mean	m/s
2	double	walkingrate_std_deviation	m/s

シミュレータ上で再現された人の歩行速度は正規分布に基づいたばらつきを持っており、その値はこの 2 つのコンフィグレーションの値で決定する。walkingrate_mean は歩行速度の平均、walkingrate_std_deviation は歩行速度の標準偏差である。

(c) localmap_width、localmap_height、localmap_scale、globalmap_scale

表 4-11　出力する地図関係のコンフィグレーション

No.	型	変数名	単位
1	int	localmap_width	cell
2	int	localmap_heigh	cell
3	double	localmap_scale	m/cell
4	double	globalmap_scale	m/cell

環境シミュレータ RTC はサービスポートを通じて 2 種類の地図を出力する。これらのコンフィグレーションはその地図についての設定を行う。lcalmap_width および localmap_height は出力する局所地図（ロボット座標系地図）の縦横の大きさを決める。また、localmap_scale は局所地図のスケール（1cell のサイズ）を決める。同様に、globalmap_scale は出力する大域地図（絶対座標系地図）のスケールを決める。

(d) following_target_id

表 4-12　人物情報のコンフィグレーション

No.	型	変数名	単位
1	int	following_target_id	-

環境シミュレータ RTC が出力する人物情報は、ロボットが特定の人物追従を行うことを想定したものとなっており、環境内のどの人物を追従対象とするのかを指定することができる。そこで、環境内に現れる何番目の人物を追従対象として指定するのかを following_target_id で決めることができる。この値は最初に出現する人物を 0 として指定する。また、追従対象を設定しない場合は負数を指定する。

(e)　robot_tread、robot_acceleration、robot_deceleration

表 4-13　ロボットの基本性能のコンフィグレーション

No.	型	変数名	単位
1	double	robot_tread	m
2	double	robot_acceleration	m
3	double	robot_deceleration	m/s2

これらのコンフィグレーションでロボットの基本的な性能を設定することができる。
robot_tread はロボットのトレッド（車輪間距離）、robot_acceleration はロボットの加速性能、そして robot_deceleration はロボットの減速性能を決定する。それぞれ想定するロボットに適した値を設定すること。

(f)　robot_init_x、robot_init_y、robot_init_heading

表 4-14　ロボット初期位置のコンフィグレーション

No.	型	変数名	単位
1	double	robot_init_x	m
2	double	robot_init_y	m
3	double	robot_init_heading	degree

これらのコンフィグレーションで環境シミュレータ RTC 起動時のロボットの初期位置を設定することができる。robot_init_x で X 座標を、robot_init_y で Y 座標を、robot_init_heading で向きをそれぞれ設定する。このときの座標は大域座標系で指定する。
大域座標系は、図 1-3を参照。

(g)　robot_move_error_x、robot_move_error_y、robot_move_error_heading

表 4-15　ロボット走行誤差に関するコンフィグレーション

No.	型	変数名	単位
1	double	robot_move_error_x	m
2	double	robot_move_error_y	m
3	double	robot_move_error_heading	degree

これらのコンフィグレーションでは､ロボットが移動したときにロボット制御命令に対して移動後の座標がどれだけずれるかを指定する。
これらのコンフィグレーションの値はそれぞれ X 座標、Y 座標、向きに対する誤差の標準偏差を表しており､ロボットが 1m 移動するごとに移動後の位置姿勢がここで指定した値に応じてずれる。
従って、これらの値を大きく設定するほどロボットは制御通り正確には進まなくなる。

(h)　range_start_position、range_end_position、range_data_num

表 4-16　レーザ距離データの計測範囲に関するコンフィグレーション

No.	型	変数名	単位
1	double	range_start_position	degree
2	double	range_end_position	degree
3	int	range_data_num	-

range_start_position および range_end_position は距離センサで計測する開始角度と終了角度を決定する。つまり、この 2 つの値によってセンサの計測範囲が決まる。

ここで角度の値は、図 1-1に示すように、センサ（ロボット）の右方向を角度の基準（0°）とすることに注意すること。

また、range_data_num は取得される距離データ数である。従って、角度分解能は

$$\frac{\text{range_end_position} - \text{range_start_position}}{\text{range_data_num}}$$ [degree] となる。

(i)　range_max_distance、range_far_code、range_std_deviation

表 4-17　レーザ距離データの計測可能距離に関するコンフィグレーション

No.	型	変数名	単位
1	double	range_max_distance	m
2	long	range_far_code	-
3	double	range_std_deviation	m

range_max_distance は距離センサで計測できる最大距離である。これを超える距離は計測結果を得ることができないものとして、計測結果の距離値に range_far_code が格納される。

つまり、range_max_distance で表される距離よりも手前に障害物が存在しない場合に、距離の値は range_far_code の値になる。

また、range_std_deviation は取得される距離データに付加される計測誤差の標準偏差である。

(j) range_peason_radius

表 4-18　レーザ距離データに映る人物に関するコンフィグレーション

No.	型	変数名	単位
1	double	range_peason_radius	m

レーザ距離データに人は円形に映る。このときの人の半径を range_peason_radius で指定することができる。また、この値を 0 にすると、距離データに人が現れないようにすることができる。

(k) range_sensor_x、range_sensor_y、range_sensor_heading

表 4-19　レーザ距離データの取り付け位置に関するコンフィグレーション

No.	型	変数名	単位
1	double	range_sensor_x	m
2	double	range_sensor_y	m
3	double	range_sensor_heading	degree

range_sensor_x および range_sensor_y は距離センサが取り付けられている位置の座標を、range_sensor_heading は角度を、それぞれ指定する。

これらの値は図 4-3に示すようなロボット中心を基準とするロボットとの相対座標で指定する。

これらのコンフィグレーションを設定することによって、ロボット中心からずれた位置に距離センサを設置した場合を想定した実験を行うことができる。

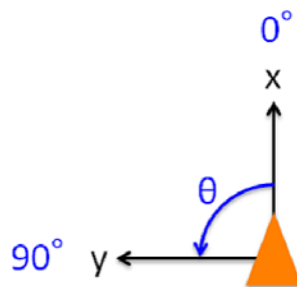



図 4-3　ロボットとの相対座標系

(l)　show_map_scale、display_range_data

表 4-20　表示に関するコンフィグレーション

No.	型	変数名	単位
1	double	show_map_scale	m/pixel
2	int	display_range_data	-

環境シミュレータRTCを起動すると、シミュレーションの様子を表示するウィンドウが現れる。show_map_scaleはこのときの画面に表示する環境（地図）のスケールを設定するためコンフィギュレーションである。また、display_range_dataに 0 以外の値を指定すると、図 4-4の様にシミュレータ上で再現されたレーザ距離データの計測した範囲を可視化して表示する。
この図で、青い線は角度毎に計測した距離を表している。

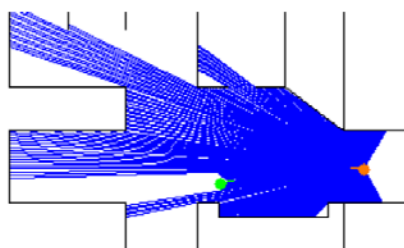



図 4-4　可視化された距離データ

(m)　simulate_people、simulate_skip

表 4-21　人物シミュレーションに関するコンフィギュレーション

No.	型	変数名	単位
1	int	simulate_people	-
2	int	simulate_skip	-

simulate_people の値を 0 に設定すると人物のシミュレーションは行われない（人が出現しない）状態でロボットのみの動作試験を行う事ができる。また、simulate_people が非ゼロの状態（人物のシミュレーションを行う状態）でもシミュレータ RTC 起動直後は人物がまだ出現していない状態で始まるが、simulate_skip を 1 に設定すると最初の人が出現したところからシミュレーションを始める。さらに、simulate_skip を 2 に設定すると、following_target_id で指定した追従対象の人物が出現するところからシミュレーションを開始することができる。

(2) 環境データ

環境シミュレータ RTC が再現する環境について記述するデータの仕様を説明する。

記述するデータは「壁（障害物）」、「目的地」、「入口」、「人の行動」および「人の初期行動」の 5 つに分類される。

要素は要素名の後ろに 1 つ、または大かっこ{}の中に複数記述する。各要素は値をコンマ「,」で区切り、セミコロン「;」で終わる。

また、「#」の後ろに続く文字列はコメントとして処理される。

記述データに対応付ける要素名は、表 4-22の通り。

表 4-22　環境データ：要素一覧

No.	記述データ	要素名
1	壁（障害物）	Walls
2	目的地	Destination
3	入口	Entrance
4	人の行動	Actions
5	人の初期行動	FirstActions

(a) 壁（障害物）

壁の情報は要素名「Walls」の後ろに 1 つ、または{}の中に複数記述する。壁は 2 つの座標を結ぶ線分として表現し、次のように記述する。

```
Walls{
     (点1のx座標[m]),(点1のy座標[m]),(点2のx座標[m]),(点2のy座標[m]);
     (点3のx座標[m]),(点3のy座標[m]),(点4のx座標[m]),(点4のy座標[m]);
}
```

図 4-5　壁（障害物）：定義例

(b) 目的地

目的地の情報は要素名「Destination」の後ろに 1 つ、または{}の中に複数記述する。

目的地は出口、地点、席、行列の 4 種類に分類され、各要素の始めに種類を表すアルファベットを記述する。

2 番目には各目的地の番号を記述する。この番号は固有のものである必要はない。

同じ種類・同じ番号の目的地が複数ある場合は、その中の最も評価の高い目的地が選ばれる。

```
Destination{
    (目的地1の定義);
    (目的地2の定義);
    (目的地3の定義);
}
```

図 4-6　目的地：定義例

① 出口

出口の情報はアルファベット「X」を先頭に、出口を表す線分を指定する 2 つの座標からなり、以下の様に記述する。

```
X,(出口番号),(点1のx座標[m]),(点1のy座標[m]),(点2のx座標[m]),(点2のy座標[m]);
```

図 4-7　出口：定義例

② 地点

地点の情報はアルファベット「P」を先頭に、座標と、その座標の何 m 以内に入れば到達したと判断するかを決める半径からなり、以下の様に記述する。

```
P,(地点番号),(x座標[m]),(y座標[m]),(半径[m]);
```

図 4-8　地点：定義例

③ 席

席の情報はアルファベット「S」を先頭に、座標を記述する。

地点との違いは、半径の指定が不要であることと、席は 1 人の人が占有するという 2 点である。つまり、他の人がすでに存在する席が目的地に設定されることはない。

席は以下の様に記述する。

```
S,(席番号),(x座標[m]),(y座標[m]);
```

図 4-9　席：定義例

④ 行列

行列の情報はアルファベット「Q」を先頭に、座標と、行列が形成される方向を表す dx、dy からなり、以下の様に記述する。

```
Q,(行列番号),(x座標[m]),(y座標[m]),(dx[m]),(dy[m]);
```

図 4-10　行列：定義例

(c) 入口

入口の情報は要素名「Entrance」の後ろに 1 つ、または{}の中に複数記述する。

入口は 2 つの座標を結ぶ線分として表現し、線分上から人が出現する。

また、何秒おきに人が出現するかという情報も入口に含まれる。入口は次のように記述する。

```
Entrance{
    (入口番号),(点1のx座標[m]),(点1のy座標[m]),(点2のx座標[m]),(点2のy座標[m]),(平均出現間隔[
秒]),(出現間隔の標準偏差[秒]);
}
```

図 4-11　入口：定義例

(d) 人の行動

人の行動の情報は要素名「Actions」の後ろに 1 つ、または{}の中に複数記述する。
行動とは、特定の目的地への移動とそこで停止する時間のことを指す。
よって、行動の情報にはその行動の番号、移動先の目的地と停止時間、さらに行動を終えた後遷移する行動の番号を含む。なお、行動の番号は固有の値でなければならない。
人の行動は次の様に記述する。

```
Actions{
    (行動番号),(目的地の種類),(目的地の番号),(次の行動),(平均停止時間[秒]),(停止時間の標準偏差[秒]);
}
```

図 4-12 人の行動：定義例

(e) 人の初期行動

人の初期行動は要素名「FirstActions」の後ろに並べて、次の様に記述する。
なお、行動番号は、人の行動（Actions）で定義した値を示す。

```
FirstActions (0番の入口から出現した人の最初の行動の番号),(1番の入口から出現した人の最初の行動の番号),...;
```

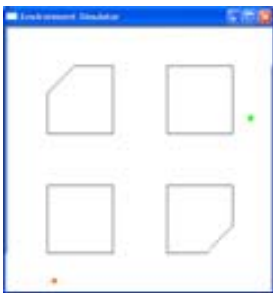



図 4-13 人の初期行動：定義例

(3) 環境データ定義例

環境シミュレータRTCでは、表 4-23に示す環境データを予め用意しており、コンポーネントの実行ファイル（exe）と共に配布している。

これらを使用することで、新規にファイルを作成しなくてもシミュレータ環境を用意することが可能である。

表 4-23　環境データ：サンプル一覧

No.	ファイル名	概要	シミュレータ表示イメージ
1	Field.txt	簡易データ	
2	Mensa.txt	学食	
3	SuperMarkt.txt	スーパーマーケット	

4.2.2. 動作セット定義ファイル

経路計画 RTC では、ロボットの動作セットを定義するファイルを使用する。

初期設定で指定されている動作セットのファイル「MotionSet.mtn」は、コンポーネントと共に配布されているが、同じく配布されるアプリケーションにより、設定、参照が可能である。

使用するロボットごとに機構、速度や加速度の制限が異なるため、それらを考慮して動作セットを作成し、利用する。

(1) 動作セット定義ツール

動作セットを定義し、その定義ファイルを作成する場合には、動作セット定義ツール（make_MotionSet.exe）を実行する。

以下のように表示されるので、出力する動作セットの定義ファイルのパスを入力する。

（注意：ここで指定したファイルが既に存在する場合は、内容が上書きされる。）

```
Please input file name to create a motion file.
_
```

図 4-14　make_MotionSet.exe 実行：動作セットの定義

以下は、ファイル名として”test.mtn”を入力した場合である。

```
test.mtn
create file : test.mtn
```

図 4-15　定義ファイルパス入力：動作セットの定義

次に、以下のように表示されるので、定義する動作の数を入力する。

```
motion num =_
```

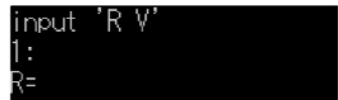



図 4-16　動作数入力：動作セットの定義

すると、以下のような表示がされる。（下図は 29 を入力した場合）

```
motion num =29
OK. motion num = 29.
```

図 4-17　動作数表示：動作セットの定義

次に、上で入力した個数分の動作を登録していく。
各動作の登録は、旋回半径を[mm]単位で入力する。
ただし、直進の場合は旋回半径に 0 を指定する。

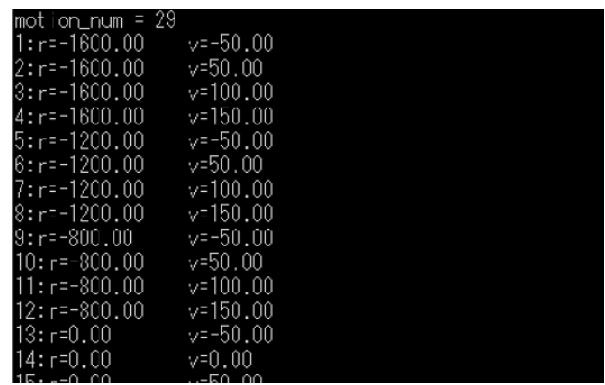
図 4-18　旋回半径入力：動作セットの定義

次に、速度の入力を求められるので、[mm/sec]単位で指定する。

V=

図 4-19　速度入力：動作セットの定義

(2)　動作セット確認ツール

動作セット定義ファイル内に定義されている動作を確認するには、動作セット確認ツール（read_MotionSet.exe）を実行する。

以下のように表示されるので、動作セット定義ファイル名のパスを入力する。

```
Please input read file name.
```

図 4-20　定義ファイルパス入力：動作セットの内容確認

すると以下のように、そのファイルに定義されている動作の数と各動作の旋回半径[mm]と速度[mm/s]が表示される。

```
motion_num = 29
1:r=-1600.00     v=-50.00
2:r=-1600.00     v=50.00
3:r=-1600.00     v=100.00
4:r=-1600.00     v=150.00
5:r=-1200.00     v=-50.00
6:r=-1200.00     v=50.00
7:r=-1200.00     v=100.00
8:r--1200.00     v-150.00
9:r=-800.00      v=-50.00
10:r= 800.00     v=50.00
11:r=-800.00     v=100.00
12:r=-800.00     v=150.00
13:r=0.00        v=-50.00
14:r=0.00        v=0.00
```

図 4-21　動作セットの表示：動作セットの内容確認

4.2.3. ステレオカメラの準備

Point Grey 社製ステレオカメラ「Bumblebee2」を PC に接続する。

(1) ステレオカメラのキャリブレーション

ステレオカメラ「Bumblebee2」で使用するキャリブレーションデータは、コンポーネント起動時に自動的に取得するため、キャリブレーションは不要である。

ステレオカメラ「Bumblebee2」は、人物検出 RTC（PeopleTrackingV2Comp）で使用する。

4.2.4. レーザ距離センサの準備

Top-URG センサを URB2.0 インタフェースで PC に接続する。接続が完了すると、レーザ距離センサはデバイス「COM*」として認識される。

デバイスマネージャよりセンサがPCに認識されたことを確認することができ、図 4-22の例では「COM3」として認識されている。

このデバイス名は、Top-URGセンサRTCのコンフィギュレーションで使用する。詳細は「6.1(1)」を参照のこと

図 4-22　デバイスマネージャの確認

4.3.　起動・終了手順

自律移動モジュール群の各コンポーネントの起動手順、および、終了手順について記述する。

4.3.1.　Windows版

(1)　起動

コンポーネントを起動すると、RT SystemEditor の Name Service View に表示される。その後、該当のコンポーネントを System Diagram にドラッグ＆ドロップすることで、他のコンポーネントとの接続が可能となる。

図 4-23は、LocalMapCmpとLocalMapViewerCmpを起動し、それぞれ接続した例である。

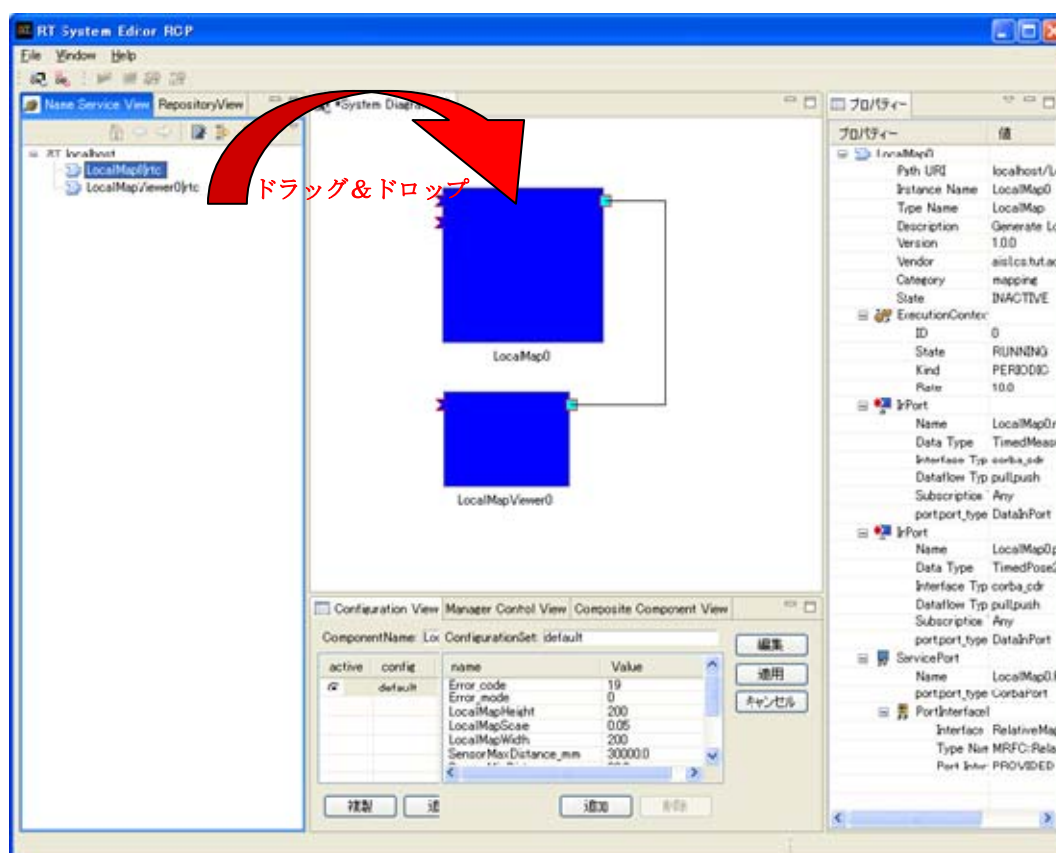



図 4-23　コンポーネントの起動

各コンポーネントは、実行ファイル（exe）をダブルクリックすることで起動するが、PeopleTrackingV2Comp では、起動時にカメラを選択する必要がある。

(a)　PeopleTrackingV2Compの起動

実行ファイル（exe）をダブルクリックし起動する。

起動すると、図 4-24のようなカメラ選択ウィンドウが表示されるので、使用するカメラを選択しOKボタンを押下する。

なお、OK ボタンを押さずにしばらく放置しておくと、コンポーネントがタイムアウトと判断して終了してしまうため、その場合には再度起動する。
選択できるカメラが表示されていない場合、カメラとPCとの接続を確認する。カメラの接続方法については、「4.2.3」を参照のこと。

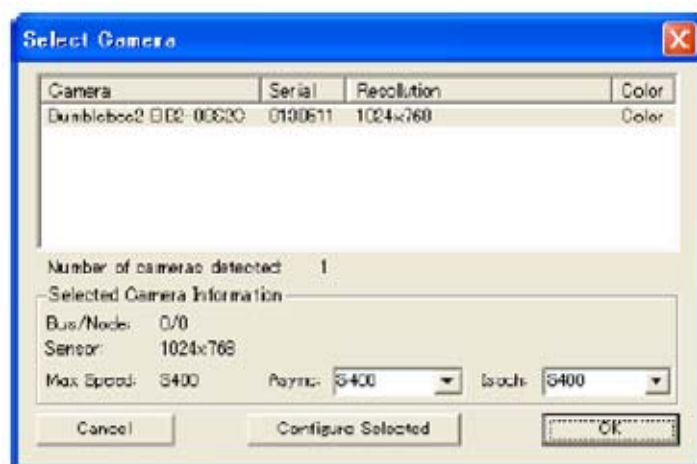



図 4-24　カメラ選択ウィンドウ

(2)　終了

起動しているコンポーネントを終了させるには、RT SystemEditor 上の終了したいコンポーネントを選択し、右クリックメニューから Exit を選択する。図では、LocalMapViewerCmp を終了する場合の例を示す。

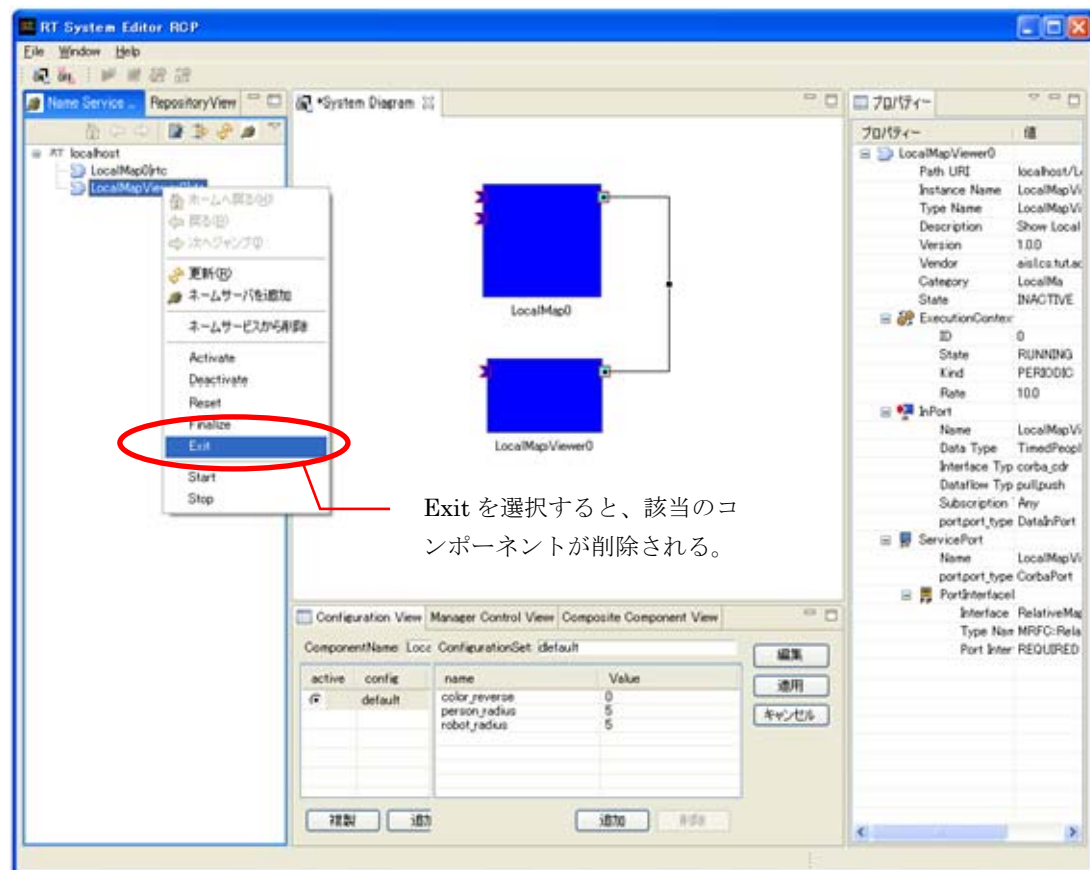



図 4-25　コンポーネントの終了

なお､各コンポーネントは､コマンドプロンプト上で Ctrl+C を入力することでも終了できる。

5　制限事項

5.1.　ロボット自己位置推定RTC

- Localization コンポーネントが読み込む地図データは、スケールが 0.1[m/cell]の大域画像にしか対応していない。その他のスケールの地図が入力された場合には処理を行わない。
- 画像データは、OpenCVが読み込むことができる形式でなくてはならない。RTCが読込可能な画像形式を表 5-1に示す。

表 5-1　読込可能な画像形式

No.	画像形式
1	Windows bitmaps (BMP, DIB)
2	JPEG files(JPEG, JPG, JPE)
3	Portable Network Graphics (PNG)
4	Portable image format (PBM, PGM, PPM)
5	Sun rasters - SR, RAS
6	TIFF files (TIFF, TIF)
7	OpenEXR HDR images (EXR)
8	JPEG 2000 images (jp2)

5.2.　MobileRobots社ロボット用制御RTC

- MobileRobotsController コンポーネントがロボットを操作するために必要なライブラリ「ARIA」は、最新版である 2.7.3（2012 年 2 月 1 日現在）では動作しないため注意すること。

5.3.　大域経路計画RTC

- 大域経路計画RTCが取得する画像データは、OpenCVが読み込むことができる形式でなくてはならない。RTCが読込可能な画像形式は、表 5-1を参照のこと。

6　付録

6.1.　コンポーネント設定データシート

(1)　コンフィギュレーションの設定

コンフィギュレーション情報はRTSystemEditor上で変更する。これはOpenRTM-aistに共通する操作であるため、OpenRTM-aistの RTSystemEditorのマニュアルを参照のこと。

なお、OpenRTM-1.0.0 では、コンフィグレーションパラメータは常時変更可能であるが、コンフィグレーションパラメータの変更は、表内の説明に記述されているタイミングでのみ反映される。

表 6-1　各コンフィギュレーション

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
1	PeopleTracking V2／PeopleTracking V2Comp	DataPortOutputType IsRobotCoord	1	値が 0 のときはデータポート TrackingPeople の位置情報の出力がカメラ座標系になり、値が 0 以外ならロボット座標系になる。変更した瞬間から出力される位置情報の座標系が切り替わる。
2		ServicePortOutputType IsRobotCoord	1	値が 0 のときはサービスポート PeopleTrackingServicePort の位置情報の出力がカメラ座標系になり、値が 0 以外ならロボット座標系になる。変更した瞬間から出力される位置情報の座標系が切り替わる。
3	TopURG_VCxx_exe／URGDataFlow CompComp	baud_rate	19200	通信速度（単位：bps)、アクティブ時、または、サービスポートのコマンドにより反映される。
4		device_name	COM1	センサ接続時に認識されたデバイス名、アクティブ時、または、サービスポートのコマンドにより反映される。
5		scan_interval	0	URG センサのスキャン間引き数、アクティブ時、または、サービスポートのコマンドにより反映される。
6		measure_mode	NORMAL	計測モード（”NORMAL”または”SHORT”)、アクティブ時、または、サービスポートのコマンドにより反映される。

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
7		start_position	0.0	計測開始位置（単位：deg） アクティブ時、または、サービスポートのコマンドにより反映される。
8		end_position	180.0	計測終了位置（単位：deg） アクティブ時、または、サービスポートのコマンドにより反映される。
9		data_grouping_number	5	まとめる方向、アクティブ時、または、サービスポートのコマンドにより反映される。
10		sensitive_mode	OFF	高感度モード（”ON”,”OFF”）。（Top-URG には高感度モードは存在しないため、”ON”であっても有効にならない） アクティブ時、または、サービスポートのコマンドにより反映される。
11		motor_slow_rate	0	モータ速度減速率 アクティブ時、または、サービスポートのコマンドにより反映される。
12	Localization／LocalizationComp	Particle	100	パーティクル数。アクティブ時に反映される。
13		debug_window	0	0 以外の値を指定するとデバッグ用のウィンドウを表示。アクティブ時に反映される。
14		USE_INIT_POSE	0	0 を指定するとパーティクルを均等に散布。 0 以外の値を指定するとパーティクルを指定した位置を中心に散布。アクティブ時に反映される。
15		INIT_X	0.0	ロボット初期 X 座標(単位：m)（USE_INIT_POSE が非 0 のときに有効）。アクティブ時に反映される。
16		INIT_Y	0.0	ロボット初期 Y 座標(単位：m)（USE_INIT_POSE が非 0 のときに有効）。アクティブ時に反映される。
17		INIT_HEADING	0.0	ロボット初期方向(単位：radian)（USE_INIT_POSE が非 0 のときに有効）。アクティブ時に反映される。

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
18	Localization／SimpleGlobalMapLoaderComp	MapData	data/gmap_png	読みこむ地図のファイル名を指定。アクティブ時に反映される。
19		InputMapScale	0.1	読みこんだ地図画像のスケール(単位：pixel)。アクティブ時に反映される。
20	SLAM／SLAMComp	Particle	50	パーティクル数。アクティブ時に反映される。
21		debug_window	0	0 以外の値を指定するとデバッグ用のウィンドウを表示。アクティブ時に反映される。
22	LocalMapRTC／LocalMapComp	LocalMapWidth	200	地図の X 方向の大きさ(単位 : cell)。アクティブ時に反映される。
23		LocalMapHeight	200	地図の Y 方向の大きさ(単位 : cell)。アクティブ時に反映される。
24		LocalMapScale	0.05	1 グリッドのサイズ(単位：m/cell)。アクティブ時に反映される。
25		SensorPositionX_m	0.0	センサ取り付け位置(単位：m)。アクティブ時に反映される。
26		SensorPositionY_m	0.0	センサ取り付け位置(単位：m)。アクティブ時に反映される。
27		SensorPositionTheta_deg	0.0	センサ取り付け角度(単位：degree)。アクティブ時に反映される。
28		SensorMaxDistance_mm	30000.0	センサの最大測定距離(単位：mm)。アクティブ時に反映される。
29		SensorMinDistance_mm	20.0	センサの最大測定距離(単位：mm)。アクティブ時に反映される。
30		Error_code	19	対象が遠すぎて値が取得できなかった際のデータ値 classicURG の場合は 19 TopURG の場合は 1 アクティブ時に反映される。
31		Error_mode	0	値に応じて URG のエラーコードの範囲を決定する。 classicURG の場合は 1 TopURG の場合は 2 アクティブ時に反映される。
32	Simulator／EnvironmentSimulatorComp	FieldDataFile	Field.txt	それぞれ、アクティブ時に反映される。 各コンフィギュレーションの詳
33		walkingrate_mean	1.33	
34		walkingrate_std_deviation	0.1	

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
35		localmap_width	200	細は、「4.2.1(1)」を参照のこと。
36		localmap_height	200	
37		localmap_scale	0.05	
38		globalmap_scale	0.1	
39		following_target_id	2	
40		robot_tread	0.3325	
41		robot_acceleration	0.400	
42		robot_deceleration	0.400	
43		robot_init_x	3.5	
44		robot_init_y	1.0	
45		robot_init_heading	200.0	
46		robot_move_error_x	0.0	
47		robot_move_error_y	0.0	
48		robot_move_error_heading	0.0	
49		range_start_position	-30.0	
50		range_end_position	210.0	
51		range_data_num	480	
52		range_max_distance	30.0	
53		range_far_code	-1	
54		range_std_deviation	0.0	
55		range_peason_radius	0.2	
56		range_sensor_x	0.0	
57		range_sensor_y	0.0	
58		range_sensor_heading	0.0	
59		show_map_scale	0.05	
60		display_range_data	0	
61		simulate_people	1	
62		simulate_skip	1	
63	PathPlanner v2.6 ver.IIS／PathPlanner V2Comp	GOAL_AREA	0.8	ロボットとゴール間の距離が指定値より小さくなったらロボットが停止する。(単位：m)。アクティブ時に反映される。
64		LOOP_TIME	0.5	経路計画のリプランニング周期の設定。(単位：秒)。アクティブ時に反映される。
65		MOVING_OBSTACLE_RADIUS	0.2	移動障害物の半径(単位：m)。アクティブ時に反映される。

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
66		ROBOT_ACCELERATION	0.4	ロボットの加速度 (単位：m/s^2)。アクティブ時に反映される。
67		ROBOT_MAX_SPEED	0.5	許容するロボットの最大速度 (単位：m/s^2)。アクティブ時に反映される。
68		ROBOT_RADIUS	0.310	ロボットの半径(単位：m) ENON, PeopleBot, PatraFourの3種類。アクティブ時に反映される。
79		TREAD	0.3325	車輪トレッドの指定(単位：m) ENON, PeopleBot, PatraFourの3種類。アクティブ時に反映される。
70		MOTION_SET_FILE	Motion Set.mtn	ロボットが使用するモーションセットの定義。アクティブ時に反映される。
71		SOUND_SET	2	デバッグ音声の設定 0：音声不使用 1：音声セット1使用 2：音声セット2使用。アクティブ時に反映される。
72		USE_WAYPOINT_PORT	2	目標地点に対して経由点を使うかどうか。 0：経由点は使わず人物追跡データを使用する。 1：経由点を使用。局所地図中で最も遠い経由点を目指す。 2：経由点を使用。各経由点の一つ一つを追従。。アクティブ時に反映される。
73		USE_WAYPOINT_HEADING	0	経由点にロボットの方向を使うかどうか 0：使わず、位置のみ。 1：使う。アクティブ時に反映される。
74		USE_LOCALIZATION_PORT	0	robot_pose データポートからロボットの現在位置を得る場合は0を指定。 localization_port データポートからロボットの現在位置を取得する場合は 1 を指定。(現在は使用していない)。アクティブ時に反映される。

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
75		USE_PEOPLE_ TRACKING	0	人物追跡を使うかどうか 0：使わない 1：使う。アクティブ時に反映される。
76		USE_VIDEO	0	local_map の結果をビデオで撮るかどうか 0：ビデオを撮らない 1：ビデオを撮る。アクティブ時に反映される。
77		USE_MAP_ SERVICE	1	map_service_port を使うかどうか 0：サービスポートは使わずデータポートを使う 1：サービスポートを使う。アクティブ時に反映される。
78		USE_PEOPLE_ SERVICE	0	people_tracking_service_port を使うかどうか 0：サービスポートは使わずデータポートを使う 1：サービスポートを使う。アクティブ時に反映される。
89	MobileRobots Controller／MobileRobots ControllerComp	com_port_no	1	ロボットと PC 間の接続の COM ポート番号。1～16 の間で指定。アクティブ時に反映される。
80		limit_speed_ meter_per_second	0.8	ロボットの制御入力で受け付ける最大速度。(単位：m/sec)。アクティブ時に反映される。
81		limit_turn_dps	90	ロボットの制御入力で受け付ける最大角速度。(単位：degree/sec)。アクティブ時に反映される。
82	GlobalPath Planner／GlobalPath PlannerComp	RobotSafetyRadius	1.0	障害物までの距離がこの値より近い点には中間目的地を生成しない(単位：m)。アクティブ時に反映される。
83		RobotVelocity	0.5	中間目的地に設定する速度(単位：m/s)。アクティブ時に反映される。
84	GlobalPath Planner／Dummy2Poses SenderComp	start_x	0.0	start_pose の X 座標(単位：m) アクティブ時に反映される。
85		start_y	0.0	start_pose の Y 座標(単位：m) アクティブ時に反映される。
86		start_heading	0.0	start_pose の向き(単位：radian) アクティブ時に反映される。

No.	ＲＴＣ名	パラメタ名	デフォルト値	説明
87		goal_x	0.0	goal_pose の X 座標(単位：m) アクティブ時に反映される。
88		goal_y	0.0	goal_pose の Y 座標(単位：m) アクティブ時に反映される。
89		goal_heading	0.0	goal_pose の向き(単位：radian) アクティブ時に反映される。
90		continuous_output	0	値が 0 のときは起動直後に 1 度だけ出力して自動的にディアクティベートする アクティブ時に反映される。
91		deactivate_to_stop	1	値が 0 以外のときはディアクティベート時に停止を意味する空のゴールを出力する アクティブ時に反映される。

6.2. メッセージ一覧

各コンポーネントにおいて出力するメッセージについて記述する。
ログの出力有無、出力形式、ログレベルは、設定ファイル（rtc.conf）の内容による。
設定ファイルの詳細は、「OpenRTM-aist デベロッパーズガイド>>RTC プログラム入門>>設定ファイル（基礎編）」を参照。

http://openrtm.org/openrtm/ja/content/%E8%A8%AD%E5%AE%9A%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB-%E5%9F%BA%E7%A4%8E%E7%B7%A8#toc3
[2012 年 2 月 1 日現在　(URL は変更される場合があります)]

6.2.1. メッセージ一覧（ShowImageComp）

出力メッセージなし。

6.2.2. メッセージ一覧（URGDataFlowCompComp）

No.	レベル	メッセージ	説明
1	ERROR	LF echo back missing.	レーザ距離センサから取得した距離データの異常。
2	ERROR	LF Command Invalid status: %s	レーザ距離センサから取得した距離データの異常。
3	ERROR	LF missing.	レーザ距離センサから取得した距離データの異常。
4	ERROR	LF decode missing.	レーザ距離センサから取得した距離データの異常。
5	ERROR	RTC_LRSServiceImpl::<<コマンド名>>: start_position: invalid value –	サービスコマンドで指定した計測開始位置のパラメータ異常。
6	ERROR	RTC_LRSServiceImpl::<<コマンド名>>：end_position: invalid value –	サービスコマンドで指定した計測終了位置のパラメータ異常。
7	ERROR	RTC_LRSServiceImpl:: <<コマンド名>>：scan_interval: invalid value –	サービスコマンドで指定したスキャン間引き数のパラメータ異常。
8	ERROR	RTC_LRSServiceImpl:: <<コマンド名>>:data_grouping_number: invalid value –	サービスコマンドで指定したまとめるステップ数のパラメータ異常。
9	ERROR	Start Measurement Error. <<ステータス>>	レーザ距離センサの計測開始失敗。
10	ERROR	<<コマンド名>> failue of powre off.	サービスコマンドによりパラメータを変更した際に、レーザ距離センサの計測停止に失敗した。
11	ERROR	<<コマンド名>> failure of start measurement.	サービスコマンドによりパラメータを変更した後に、レーザ距離センサの計測開始に失敗した。
12	ERROR	resetSensor failue of reset sensor.	サービスコマンド（リセット）の際に、レーザ距離センサのパラメータリセットに失敗したことを示す。
13	ERROR	resetSensor failure of set motor slow rate.	サービスコマンド（リセット）の際に、レーザ距離センサのモータ回転速度の設定に失敗した

No.	レベル	メッセージ	説明
			ことを示す。
14	ERROR	resetSensor failure of sensitive mode.	サービスコマンド（リセット）の際に、レーザ距離センサの計測モードの設定に失敗したことを示す。
15	ERROR	checkAllConfigureParameters error <<メッセージ>>	コンフィギュレーションパラメータの設定項目異常。
16	ERROR	getResponse Port closed.	RTC の制御異常、センサへアクセスしたがセンサのポートがオープンしていない。
17	ERROR	onActivated parameter error.	アクティブ化時のコンフィギュレーションパラメータの異常を示す。
18	ERROR	onActivated failure of reset sensor.	アクティブ化時に、レーザ距離センサのリセットに失敗したことを示す。
19	ERROR	onActivated failure of set motor slow rate.	アクティブ化時に、レーザ距離センサのモータ回転速度の設定に失敗したことを示す。
20	ERROR	onActivated failure of sensitive mode.	アクティブ化時に、レーザ距離センサの計測モードの設定に失敗したことを示す。
21	ERROR	onActivated failure of start measurement.	アクティブ化時に、レーザ距離センサの計測開始に失敗したことを示す。
22	WARN	characterDecode: length: %ld - mode: %ld - Length error	レーザ距離センサより取得した距離データの異常を示す。
23	WARN	parseVV Default value used.	レーザ距離センサより取得したレーザ距離センサの計測モードの異常を示す。
24	WARN	parsePP Default value used.	レーザ距離センサよりバージョン情報の取得に失敗したことを示す。
25	WARN	createMDMS Invalid value: %s. Default value: NORMAL used.	レーザ距離センサよりパラメータ情報の取得に失敗したことを示す。
26	WARN	createMDMS Invalid value: %ld. Default value: 44 used.	計測開始位置の設定において、不正なパラメータが指定されたためデフォルト値を適用したことを示す。
27	WARN	createMDMS Invalid value: %ld. Default value: 725 used.	計測終了位置の設定において、不正なパラメータが指定されたためデフォルト値を適用したことを示す。
28	WARN	createMDMS Invalid value: %ld. Default value: 5 used.	まとめるステップ数の設定において、不正なパラメータが指定されたためデフォルト値を適用したことを示す。
29	WARN	createSS Invalid value: %ld. Default value: 19200 used.	ボーレートの設定において、不正なパラメータが指定されたためデフォルト値を適用したことを示す。
30	WARN	createCR Invalid value: %ld. Default value: 00 used.	モータ回転速度の設定において、不正なパラメータが指定されたためデフォルト値を適用したことを示す。
31	WARN	createHS Invalid value: %s. Default value: OFF used.	計測モードの設定において、不正なパラメータが指定されたためデフォルト値を適用したこと

No.	レベル	メッセージ	説明
			を示す。
32	WARN	getVersion Command VV no response.	レーザ距離センサより、バージョン情報取得コマンドの応答が一定時間経っても得られなかったことを示す。
33	WARN	getParam Command PP no response.	レーザ距離センサより、センサパラメータ情報取得コマンドの応答が一定時間経っても得られなかったことを示す。
34	WARN	resetParam Command RS no response.	レーザ距離センサより、リセットコマンドの応答が一定時間経っても得られなかったことを示す。
35	WARN	setMotorSlowRate Command CR no response.	レーザ距離センサより、モータ回転速度変更コマンドの応答が一定時間経っても得られなかったことを示す。
36	WARN	setSensitiveMode Command HS no response.	レーザ距離センサより、計測モード切替コマンドの応答が一定時間経っても得られなかったことを示す。
37	WARN	startMeasurement Command MD or MS no response.	レーザ距離センサより、計測開始要求の応答が一定時間経っても得られなかったことを示す。
38	WARN	powerOff Command QT no response.	レーザ距離センサより、計測停止要求の応答が一定時間経っても得られなかったことを示す。
39	WARN	onFilelize called. receiver() exit	計測を停止せず RTC を終了し、強制的に計測を停止したことを示す。
40	WARN	receiver Status error.	レーザ距離センサより予期せぬ不正な情報を受信したことを示す。
41	WARN	getResponse Response buffer overflow.	レーザ距離センサより取得した距離情報の長さが不正(10KB 以上)を超えたことを示す。 ※この場合取得データは無視する。
42	INFO	Measurement starts.	レーザ距離センサへ計測開始コマンドを送信したことを示す。
43	INFO	Sensor ON	レーザ距離センサへ計測開始コマンドを送信したことを示す。
44	INFO	Sensor OFF	レーザ距離センサへ計測停止コマンドを送信したことを示す。
45	INFO	Parameters for mesurement reset.	レーザ距離センサへリセット計測開始コマンドを送信したことを示す。
46	INFO	Service: resetSensor	サービスコマンド（センサリセット）が実行されたことを示す。
47	INFO	Service: getLatestData	サービスコマンド（最新距離データの取得）が実行されたことを示す。
48	INFO	Service: getLatestData exit	サービスコマンド（最新距離データの取得）が終了したことを示す。
49	INFO	Service: getStatus	サービスコマンド（ステータス情報の取得）が実行したことを示す。

No.	レベル	メッセージ	説明
50	INFO	Service: getStatus exit	サービスコマンド（ステータス情報の取得）が終了したことを示す。
51	INFO	SerialPort Connected to [<<ポート番号>>].	レーザ距離センサが接続するシリアルポートに接続したことを示す。
52	INFO	SerialPort Disconnected.	レーザ距離センサが接続するシリアルポートと接続を切断したことを示す。
53	INFO	setBaudRate BaudRate OK.	ボーレートの設定に成功したことを示す。
54	INFO	setMotorSlowRate Motor slow rate OK.	モータ回転速度の設定に成功したことを示す。
55	INFO	setSensitiveMode Sensitive mode OK.	計測モードの切り替えに成功したことを示す。
56	INFO	startMeasurement Measurement starts.	レーザ距離センサの計測開始に成功したことを示す。
57	INFO	Power off.	レーザ距離センサの計測停止に成功したことを示す。
58	INFO	receiver Command accepted.	レーザ距離センサより計測開始の受付が返却されたことを示す。
59	INFO	setParams	計測パラメータの変更が行われたことを示す。
60	INFO	Start position:<<設定値>>	計測パラメータ(計測開始位置)の変更が行われたことを示す。
61	INFO	End position: <<設定値>>	計測パラメータ(計測終了位置)の変更が行われたことを示す。
62	INFO	Scan interval: <<設定値>>	計測パラメータ(スキャン間引き数)の変更が行われたことを示す。
63	INFO	Data grouping number: <<設定値>>	計測パラメータ(まとめるステップ数)の変更が行われたことを示す。
64	INFO	Device Already Opened.	RTC がレーザ距離センサとの接続を試み、既に接続されていたことを示す。
65	INFO	onInitialize Component created.	onInitialize が実行され、RTC が生成されたことを示す。
66	INFO	onFinalize Finalizing...	onFinalize が実行され、RTC の破棄が開始されたことを示す。
67	INFO	onFinalize Component finalized.	onFinalize が実行され、RTC が破棄されたことを示す。

6.2.3. メッセージ一覧（PeopleTrackingV2Comp）

No.	レベル	メッセージ	説明
1	ERROR	flycaptureStart() reported The requested bandwidth would exceed the maximum.	IEEE1394 のドライバに問題があり、帯域速度が不十分であることを示す。Bumblebee2 のマニュアル、もしくは ViewPLUS の Web サイト（http://www.viewplus.co.jp/support/?p=87）を参考にパッチ fixSP2g.exe を当てる。
2	INFO	Calibration File successfully saved at	カメラからパラメータを取得し、ファイルに保

		bumblebee1234567.cal	存したことを示す。数字はカメラのシリアル番号。
3	INFO	Get Context from bumblebee1234567.cal	ファイルからカメラパラメータを取得したことを示す。ファイルが存在しない場合には自動的にカメラから取得する。

6.2.4.　メッセージ一覧（LocalizationComp）

No.	レベル	メッセージ	説明
1	ERROR	Service port is not conected with service provider,.	プロバイダーポートが接続されていない異常
2	INFO	Waiting input initial pose.	入力データ待ち。
3	INFO	Input Robot Pose	ロボット情報が入力されたことを示す。
4	INFO	mao size = (<<値>>, <<値>>)	地図のサイズを示す。
5	INFO	Step : <<値>>	ステップ数を示す。
6	INFO	Start to predict and to update Global Map	予測開始、地図更新の開始を示す。
7	INFO	Finish to predict and to update Global Map	予測開始、地図更新の完了を示す。
8	INFO	Start to calculate Likelihoods	尤度計算処理開始を示す。
9	INFO	Finish to calculate Likelihoods	尤度計算処理終了を示す。
10	INFO	Start resampling process	りサンプリングの開始を示す。
11	INFO	Finish resampling process	りサンプリングの完了を示す。
12	INFO	Finish one sycle processing	1 サイクルの完了を示す。
13	INFO	Output Estimated Robot Pose (<<値>> , <<値>>, <<値>>)	ロボットの推定位置の出力を示す。
14	INFO	Input LRF Data	LRF データの入力を示す。
15	INFO	The end of Localization	コンポーネントの終了を示す。
16	INFO	Total Execution Time : <<値>> [s]	トータル実行時間を示す。
17	INFO	Required Time per cycle Average :<<値>> [ms/cycle] Maximum : <<値>>[ms] Minimum : <<値>>[ms]	1 サイクル辺りの実行時間を示す。

6.2.5.　メッセージ一覧（SimpleGlobalMapLoaderComp）

出力メッセージなし。

6.2.6.　メッセージ一覧（SLAMComp）

No.	レベル	メッセージ	説明
1	INFO	Output Estimated Robot Pose (<<値>>, <<値>>, <<値>>)	ロボットの推定位置を示す。
2	INFO	Finish to calculate Likelihoods	尤度計算処理終了を示す。
3	INFO	Finish one cycle processing.	1 サイクル処理終了を示す。
4	INFO	Start to make GlobalMap	地図生成の開始を示す。

5	INFO	Finish to make GlobalMap	地図生成の終了を示す。
6	INFO	Input Robote Pose	ロボット位置の入力を示す。
7	INFO	Input LRF Data	LRF データの入力を示す。
8	INFO	Step : <<値>>	処理ステップ数を示す。
9	INFO	Start to predict and to update GlobalMap	予測と地図データの更新開始を示す。
10	INFO	Finish to predict and to update GlobalMap	予測と地図データの更新終了を示す。
11	INFO	The end of SLAM	コンポーネント終了を示す。
12	INFO	Total Execution Time : <<値>> [s]	トータル実行時間を示す。
13	INFO	Required Time per cycle Average :<<値>> [ms/cycle] Maximum : <<値>>[ms] Minimum : <<値>>[ms]	1 サイクル辺りの実行時間を示す。

6.2.7. メッセージ一覧（LocalMapComp）

出力メッセージなし。

6.2.8. メッセージ一覧（EnvironmentSimulatorComp）

No.	レベル	メッセージ	説明
1	ERROR	Component create failed.	コンポーネントの生成に失敗。
2	WARN	Start to build spatial network Number of generated Subareas =<<値>> Number of optimised Subareas =<<値>>	コンフィギュレーションで指定された環境データに入口が無い場合、デフォルトの環境データで構築したことを示す。
3	INFO	Start Initialization	アクティブ化時に、初期化が開始されたことを示す。
4	INFO	Finish Initialization	アクティブ化時に、初期化が終了されたことを示す。
5	INFO	The end of Simulation. Total Execution Time :<<値>>[s] Required Time per cycle Average :<<値>> [ms/cycle] Maximum : <<値>>[ms] Minimum : <<値>>[ms]	シミュレータの終了。トータル実行時間と 1 サイクル辺りの所要時間を示す。
6	INFO	[STEP　<<値>>]	ステップ数を示す。
7	INFO	Number of persons = <<値>>	人間の数を示す。
8	INFO	Triangules are merged into convex polygons.	自由線分で接する多角形を凸型の範囲で統合したデータを加えたことを示す。

6.2.9. メッセージ一覧（PathPlannerV2Comp）

No.	レベル	メッセージ	説明

1	WARN	danger!!!:COLLISION	衝突を示す。
2	WARN	Dilation2:Something wrong・・・・・Possibility of collision	何かと衝突した可能性があることを示す。
3	INFO	PathPlanner: onInitialize	コンポーネントの初期化処理を示す。
4	INFO	Run Frequency: <<値>>	実行周期を示す。
5	INFO	Robot is here ： x=<<値>> , y=<<値>> , direction=<<値>>	ロボット開始位置を示す。
6	INFO	path predict x=<<値>> y=<<値>> theta=<<値>>	経路予測を示す。
7	INFO	onExecute Start step=<<値>> call period t :<<値>> [sec]	onExecute の実行。呼び出された回数、呼び出された周期を示す。
8	INFO	Get a local map state time ： <<値>>[msec]	局所地図のステータス情報を読み込むのにかかった時間を示す。
9	INFO	PathPlanner : <<値>>	コンポーネントの状態を示す。
10	INFO	Connected to PeopleBot	PeopleBot への接続完了を示す。
11	INFO	Number of motion set = <<値>>	動作セット番号を示す。
12	INFO	Goal Position x=<<値>> y=<<値>>	目的地座標を示す。
13	INFO	Best-first search start route	最適な経路探索の開始を示す。
14	INFO	BEST PATH and REUSE	最適な経路と再利用を示す。
15	INFO	Time for making Best Tree = <<値>>[msec]	最良優先探索による経路計画の時間を示す。
16	INFO	node for best path <<値>>	最適な経路が引かれる地点を示す。
17	INFO	Route Search Computation time : <<値>> [msec]	経路探索の計測時間を示す。
18	INFO	Build Path Start, closest node=<<値>>	経路探索開始時の最も近い地点を示す。
19	INFO	build path : <<値>> node	経路探索に用いた地点を示す。
20	INFO	Memory consumption:<<値>> MByte	メモリ消費量を示す。
21	INFO	calculation period for this step : <<値>> [msec]	該当処理の計測時間を示す。
22	INFO	Total time : <<値>> msec	トータル実行時間を示す。
23	INFO	Total computation time : msec	トータル計算時間を示す。
24	INFO	An average computation time of the loop : <<値>> msec	平均計算時間を示す。
25	INFO	The maximum cimputation time :: <<値>> msec	最大計算時間を示す。

6.2.10. メッセージ一覧（MobileRobotsControllerComp）

No.	レベル	メッセージ	説明
1	ERROR	Syncing <<値>> No packet.	ロボットからの通信がない異常
2	ERROR	Could not connect, no robot Responding.	ロボットに接続できない異常。

3	ERROR	Failed to connect to robot.	ロボットとの接続に失敗。
4	ERROR	Failed to connect.	接続の失敗。
5	ERROR	asyncConnect failed because robot is not running in its own thread.	ロボットが起動していない異常。
6	WARN	Robot may be connected but not open, trying to dislodge.	ロボットを移動させると接続が抜ける場合があります。
7	WARN	Trying to close possible old Connection.	古い接続のクローズを試みることを示す。
8	INFO	COM:<<値>>	COM ポートのデバイス名を示す。

6.2.11. メッセージ一覧（Dumy_velocity_dataComp）

出力メッセージなし。

6.2.12. メッセージ一覧（GlobalPathPlanner）

No.	レベル	メッセージ	説明
1	ERROR	Service provider is not activated.	サービスプロバイダーが非アクティブである異常。
2	WARN	Gemerated no waypoints (Coordinate is out of area)	座標が領域外であり、経路点を生成できないことを示す。
3	INFO	Compute Global Path from (<<値>>, <<値>>) to (<<値>>, <<値>>)	2 点の座標値の経路計算を示す。
4	INFO	Number of Generated waypoints = <<値>>	生成された経路点の数を示す。

6.2.13. メッセージ一覧（Dummy2PosesSenderComp）

No.	レベル	メッセージ	説明
1	ERROR	Component create failed.	コンポーネントの生成に失敗したことを示す。

6.3. トラブルシューティング

(1) ネームサーバが起動しない

ネームサーバのコンソール画面が開かないケースがあり、この場合、以下のような原因が考えられる。

(a) omniORBpyがインストールされていない

Open-rtm-aist 1.0.0(C++版)には omniORB が含まれているが、カスタムインストールを選択すると、omniORB をインストールせずに OpenRTM-aist をインストールすることもできる。

omniORB が入っていない場合も考えられるので、omniORB がインストールされているか確認すること。

(b) 環境変数OMNI_ROOTが設定されていない

「Start Naming Service」は“%RTM_ROOT%¥bin¥rtm-naming.bat”にあるバッチファイルからネームサーバ(omniNames.exe)を起動する。この際、“omniNames.exe”を参照するために環境変数 OMNI_NAMES を利用している。通常インストーラで OpenRTM-aist をインストールした場合には、OMNI_ROOT 環境変数が自動で設定されるが、何らかの理由で環境変数が無効になっていたり、手動でインストールした場合などは、環境変数が設定されていないことがある。

環境変数 OMNI_ROOT が設定されていることを確認すること。環境編巣は、

- 「コントロールパネル」>>「システム」>>「詳細設定」タブ>>「環境変数」
- 「マイコンピュータ」を右クリック、「プロパティ」を選択、「詳細設定」タブ>>「環境変数」などから参照・編集することができる。

(c) その他

ユーザ名が 2 バイト文字の場合、ログを出力するフォルダを適切に設定できずに“omniNames.exe”の起動に失敗する場合がある。その場合、環境変数 TEMP を 2 バイト文字を含まない場所に設定することで改善する場合がある。適当なテンポラリディレクトリ（以下のケースでは C:¥temp）を作成し、そこを環境変数“TEMP”が指すように、“rtm-naming.bat”の先頭部分で以下のように設定する。

```
set cosnames="omninames"
set orb="omniORB"
set port=%1
rem set OMNIORB_USEHOSTNAME=localhost
set PATH=%PATH%;%OMNI_ROOT%¥bin¥x86_win32
set TEMP=C:¥temp
```

また、稀なケースだが、ホスト名やアドレスの設定の問題で、起動が上手くいかないケースがある。その場合、利用している PC の IP アドレスを“omniNames.exe”に教えてあげると上手

くいくケースがある。環境変数“OMNIORB_USEHOSTNAME”を以下のように設定する。（以下は、自ホストの IP アドレスが 192.168.0.11 の場合の例）。

```
set cosnames="omninames"
set orb="omniORB"
set port=%1
set OMNIORB_USEHOSTNAME=192.168.0.11
set PATH=%PATH%;%OMNI_ROOT%¥bin¥x86_win32
```

(2)　人物検出RTC

(a)　カメラ選択ダイアログにカメラが表示されない

ステレオカメラが正しく接続されていない場合、カメラ選択画面に接続中のカメラが表示されない。ステレオカメラの接続を確認すること。

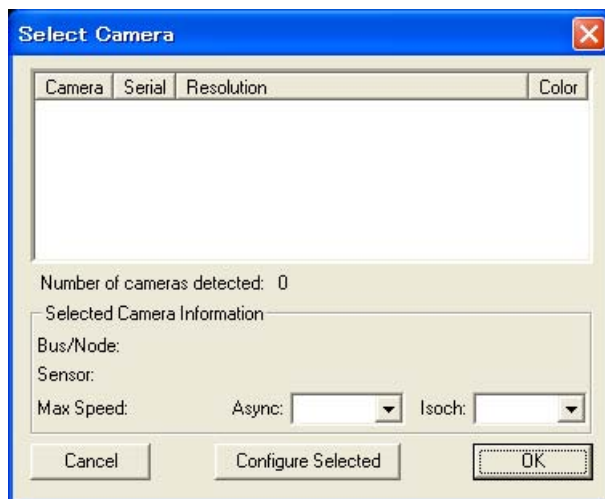



図 6-1　カメラが表示されない場合

(3) 環境シミュレータRTC上でロボットが移動しない

環境シミュレータ RTC 上で、仮想ロボットが動作しない場合、以下の原因が考えられる。

(a) RTCが正しく接続されていない

各 RTC が正しく接続されていない可能性があります。接続関係を確認すること。

(b) コンフィギュレーションが正しく設定されていない

環境シミュレータ RTC を経路計画コンポーネント（PathPlannerV2Comp）と組み合わせて使用し、人物追従を行う場合、経路計画コンポーネントのコンフィギュレーションが以下のように設定されている必要がある。

```
USE_PEOPLE_TRACKING    ：1（人物追跡を使う）
USE_WAYPOINT_PORT      ：0（経由点は使わず人物追跡データを使用する）
```

(4) Top-URGセンサRTC

(a) デバイス使用中のエラーが発生する

センサ RTC が異常終了し、デバイスのクローズ処理が正常に行われなかった場合、デバイスが使用中である情報が残ったままとなる。

この情報により、次回センサ RTC を実行し、アクティブ状態への遷移処理を行う場合、デバイスが使用中であるというエラーが発生する。

本エラーが発生した場合、エラー状態になったセンサ RTC をリセット（RT SytemEditor より操作）することで非アクティブ状態に復帰させ、再度アクティブ状態へ遷移させることを試みることで解決する。

(b) センサRTCがアクティブ状態にならない

センサ RTC のコンフィギュレーションの値が未対応である場合に、本現象が発生する。

このとき、各センサ RTC は、エラー状態になっており、RT SytemEditor において RTC の reset を行い、非アクティブ状態に戻す。そして、コンフィギュレーションの値を正しい値に設定した後、再度アクティブ状態へ遷移させる。

(c) データポートから出力されるデータが変化しない

センサ RTC は、周期的にセンサから距離データを取得しているため、出力用ポートから出力されるデータには変化があるはずである。しかし、この変化がなくなった場合、センサに異常が発生し、センサが応答しなくなっていることが考えられる。

このような場合、センサ RTC を終了させ、センサ本体の電源を入れ直し、再度センサ RTC の実行を試みる。